
NETTITUTORIAALIEN TUOTANTO-OHJELMAN SUUNNITTELU JA TOTEUTUS




Ammattikorkeakoulun opinnäytetyö

Mediatekniikan koulutusohjelma

Riihimäki, 21.11.2011

Tommi Tuominen



HAMK Riihimäki
Mediatekniikan koulutusohjelma
Ohjelmointi

Tekijä	Tommi Tuominen	Vuosi 2011
Työn nimi	Nettitutoriaalien tuotanto-ohjelman suunnittelu ja toteutus	

TIIVISTELMÄ

Erilaisten sähköisten ohjeistusten ja oppaiden tarve on lisääntynyt. Opinnäytetyön aiheena on tuoteohjeistusten toteuttamiseen tarkoitetun web-pohjaisen sovelluksen tekninen suunnittelu ja ohjelmointi toimeksiantajana mainostoimisto Precis. Opinnäytetyönä valmistui nettitutoriaalien hallintaohjelma sisällöntuottajien tarpeisiin. Tavoitteena oli, että tekninen ja visuaalinen toteutus yhdessä takaisivat sen monipuolisen käytettävyyden. Tavoitteena oli myös täydentää mainostoimiston tarjoamia verkkopalveluja asiakasyrityksille.

Nettitutoriaalien hallintaohjelma suunniteltiin tukeutuen Jacob Nielsenin (1993) käytettävyysteoriaan, josta käytettävyyden ja hyödyllisyyden lisäksi tärkeimpinä kriteereinä toimivat opittavuus, tehokkuus, miellyttävyys, muistettavuus ja virheettömyys. Ohjelman suunnittelu ja toteutus tapahtuivat erilaisten kokeilujen ja testauksen avulla. Tätä työtapaa kutsutaan tekniseksi suunnitteluksi. Ohjelmoinnin apuna käytettiin internetistä löytyviä Php ja MySQL –tutoriaaleja.

Nettitutoriaalien hallintaohjelma vastaa pääosin Jakob Nielsenin käytettävyyssääntöjä helposti opittavalla ja miellyttävän, selkeän ulkoasun omaavalla käyttöliittymällä. Ohjelman täyttää tehokkuutta ei kuitenkaan ole testattu johtuen oikeiden tutoriaalikohteiden ja Php-serverin puutteesta. Yleisesti sisällöntuotto-sovelluksiin verrattuna kehitetyssä ohjelmassa on eroja, jotka voivat täydentää mainostoimiston verkkopalveluja. Ohjelma esimerkiksi luo jokaisesta tutoriaalista oman kokonaisuutensa, joka voidaan helposti jakaa kansiona asiakkaalle.

Jatkokehityskohteina voisivat olla valmiit asiakaskohtaiset tutoriaalirungot, jotka mahdollistaisivat automaattisen sivumuotoilun ja sisältöelementtien paikat asiakkaan ennalta määrittämien arvojen mukaisesti. Jatkotutkimuksella saataisiin selville myös asiakaskäyttäjien mielipiteitä ja parannusehdotuksia, jolloin nettitutoriaali-ohjelmasta voitaisiin kehittää vielä laajemmin käyttökelpoinen.

Avainsanat internet, opas, hallintaohjelma, verkko-ohjelmointi, tekninen suunnittelu

Sivut 40 s.

HAMK Riihimäki
Degree Programme in Media Technology
Programming

Author	Tommi Tuominen	Year 2011
Subject of Bachelor's thesis	An online tutorial management application – design and programming	

ABSTRACT

The demand for different kinds of electronic tutorials has grown. The subject of this thesis is the engineering design, including programming of an online tutorial management application for content producers, commissioned by an advertising agency entitled Precis. The goal of the project was to integrate technical and visual execution of the application for maximal usability in various contexts. Another goal was to expand the product range with an online tutorial for clients of the advertising agencies.

The software design process is based on Jakob Nielsen's (1993) guidelines for usability. The most important guidelines are learnability, efficiency, pleasantness, remembering and faultlessness. The programming process was carried out experimenting with multiple approaches and testing. Various PHP and MySQL tutorials were used as references in programming.

The online tutorial management application honors the usability guidelines of Jakob Nielsen. The application features an easy-to-understand user interface, which is easy to learn, and it is visually pleasant. The full efficiency of the application has not been tested, since it has not been possible to test in a real-life user context or on a real web-server either. In the application, there are some key differences compared to other website content-management applications. For example, the tutorials created with the online tutorial management application can be easily sent to the client in a folder.

Further development of the application should be done, for example, creation of pre-defined tutorial templates for different customers. Pre-defined templates would allow better control over placing and positioning of content elements. Opinions and improvements should be gathered from client users for further researching to make the application even more usable.

Keywords internet, tutorial management application, online programming, engineering design

Pages 40 p.

SISÄLLYS

1	JOHDANTO.....	1
1.1	Precis-mainostoimisto ja sen nettiympäristö.....	1
1.2	Sähköiset ohjeistukset mainostoimistossa.....	1
2	OHJELMAN KÄYTETTÄVYYDESTÄ.....	3
3	NETTITUTORIAALIOHJELMAN SUUNNITTELU	6
3.1	Sovelluksen toimintaympäristö.....	6
3.2	Sovellushierarkia ja toiminta.....	6
3.2.1	Valmiin sovelluksen ominaisuudet.....	7
3.2.2	MySQL-tietokannan rakenne.....	7
3.3	Toteutuksessa tarvittavat tekniikat	8
3.3.1	Html, Css, JavaScript	8
3.3.2	Php, MySQL.....	8
3.4	Käyttöliittymän suunnittelu.....	9
4	NETTITUTORIAALIOHJELMAN TOTEUTUS	12
4.1	Toteutuksessa käytetyt sovellukset	12
4.2	Kehityksen kulku	12
4.2.1	Html-runko	14
4.2.2	JavaScript ja Php –toiminnot.....	15
4.3	Sovelluksen testaus	17
5	NETTITUTORIAALIOHJELMAN OMINAISUUDET	18
5.1	Tutoriaalin hallinta	18
5.1.1	Tutoriaalin luonti	19
5.1.2	Tutoriaalin avaus	21
5.1.3	Tutoriaalin poisto.....	23
5.2	Sivujen hallinta.....	23
5.2.1	Sivun luonti	24
5.2.2	Sivun otsikon vaihto	25
5.2.3	Sivun poisto	25
5.3	Teksti-, kuva- ja videotyökalut.....	27
5.3.1	Elementtien ominaisuudet	29
5.3.2	Elementin lisäys.....	30
5.3.3	Elementin tietojen muuttaminen.....	31
5.3.4	Elementin poisto	31
5.4	Sivuston tyylitiedoston asetukset	32
5.5	Luotu tutoriaalisivu	32
6	YHTEENVETO JA POHDINTAA.....	33
6.1	Nettitutoriaaliohjelman suunnittelu ja toteutus	33
6.2	Toimintaympäristö	33
6.3	Toteutuksessa ilmenneet ongelmat.....	34
6.4	Oppina itselle.....	34
6.5	Jatkotutkimus ja kehittäminen.....	34
	LÄHTEET	36

1 JOHDANTO

Erilaisten sähköisten ohjeistusten tarve on lisääntynyt, kun niitä on saatavilla verkon kautta mobiilistikin, esimerkiksi autokorjaamoissa tai metsätöissä. Opinnäytetyön aihe on ohjeistusten tuottamiseen tarkoitetun web-pohjaisen sovelluksen suunnittelu ja toteutus. Tässä raportissa kuvataan mainostoimiston nettiympäristöä, sähköisten web-pohjaisten tutoriaalisivujen tarkoitusta ja käydään läpi sovelluksen kehityksen kulku suunnittelusta valmiiksi asti. Lopuksi arvioidaan toteutuksen toimivuutta.

1.1 Precis-mainostoimisto ja sen nettiympäristö

Opinnäytetyö suoritettiin hämeenlinnalaisessa Precis-mainostoimistossa (<http://www.precis.fi/ullakko/>). Se on perustettu vuonna 1994. Yrityksen palveluihin kuuluvat muun muassa lehtimainosten, julisteiden, esitteiden ja muun painetun mainosmateriaalin suunnittelu ja tuotanto yrityksille. Viime vuosina yritys on laajentanut toimintaansa videomainontaan. Precis tekee myös erilaisia suunnittelutöitä, kuten nettisivujen ulkoasuja.

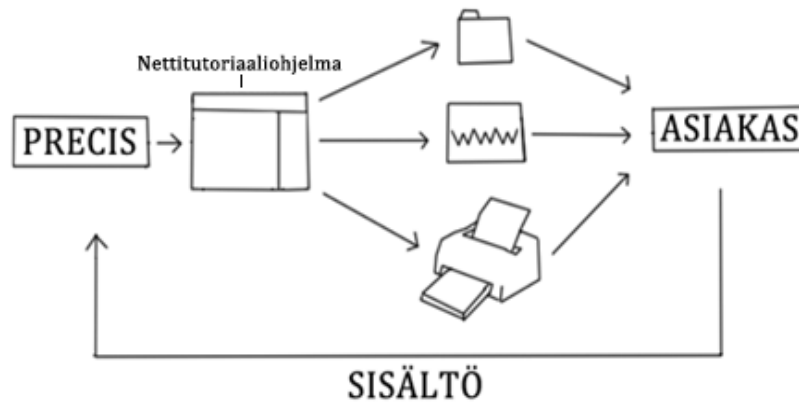
Mainostoimiston asiakkaita ovat yritykset, jotka tarvitsevat näyttäviä mainoksia tai messurekvisiittia. Nimekkäimpiä asiakkaita ovat esimerkiksi Kone Cranes, Ruukki ja Subaru. Preciksessä työskentelee yhteensä 12 vakituista työntekijää ja kaksi freelancer-työntekijää tämän opinnäytetyön toteutusvuonna 2011.

Preciksellä on oma tiedostonjako-, ftp-, ja sähköpostipalvelin. Monet Preciksen suunnittelemista web-sivustoista ovat Opiferumin (<http://www.opiferum.com/>) tekemiä ja siten Opiferumin omilla palvelimilla. Luultavasti myös tässä asiakirjassa käsiteltävä web-pohjainen tutoriaalien tuotanto-ohjelma tulee olemaan Opiferumin palvelimella.

1.2 Sähköiset ohjeistukset mainostoimistossa

Preciksellä on aiemmin tehty ohjeistuksia pdf- ja paperimuodossa. Yritys kuitenkin halusi ohjeistuksille formaatin, joka toimii kaikilla laitteilla missä ja milloin vain. Päätyminen nettipohjaisiin tutoriaalisivuihin oli siis lähes ainoa vaihtoehto, koska nettisivut toimivat lähes kaikissa laitteissa, joissa on nettiyhteys.

Ohjeistuksia tarvitaan asiakkaan tuotteille ja niiden käyttöönottoon tai asennukseen. Preciksen yksi merkittävimmistä asiakkaista on Oilon, joka on kehittyvä ympäristö- ja energiatekniikan perheyritys. Esimerkkinä voidaan siis ajatella lämpöpumpun asennusta ja siinä tarvittavaa ohjeistusta. Oilonilta voidaan antaa kätevästi linkki asennusohjesivustoon, jonka asentaja voi tulostaa tai katsoa esimerkiksi mobiililla web-päätteellä paikanpäällä. Sisällön ohjeistukseen tuottavat siis yritys ja mainostoimisto yhdessä kuten tähänkin asti.



Kuva 1 Nettitutoriaali mainostoimiston lisäpalveluna.

Web-pohjainen ohjeistusalusta/ järjestelmä voi olla siis yksi mainostoimistojen lisäpalveluista yrityksille nykyisessä verkkoympäristössä. Ohjeistusten kehitys on kulkeutunut paperilta esite- ja opasmuodosta, pdf-dokumenttien kautta tutoriaaleiksi verkkoon. Kaikkia formaatteja käytetään edelleen myös rinnakkain (ks. Kuva 1).

Ajatus kyseiseen web-pohjaiseen ohjeistusjärjestelmään tuli ollessani työharjoittelussa Preciksessä kun kysyin yrityksen toimitusjohtajalta mahdollisia opinnäytetyöideoita. Vaikka työnkuvaani yrityksessä kuului pääasiassa 3d-mallien luonti Modo 3d -ohjelmalla, tuntui ohjelmointiprojekti mielenkiintoiselta haasteelta: ohjelma piti suunnitella ja kehittää alusta lähtien erilaisten mainostoimiston asiakasyritysten ja niiden tuotteiden ohjeistustarpeisiin.

Suurin projektia ohjaava kysymys oli se, miten nettitutoriaali-ohjelman tekninen ja visuaalinen toteutus yhdessä takaisivat sen monipuolisen käytettävyyden, siihen sisältöä tuottavien mainostoimiston ammattilaisten keskuudessa.

Jatkossa tässä raportissa käytän opinnäytetyöstä siis hallintaohjelmasta, nimitystä *nettitutoriaali-ohjelma* erotuksena valmiista sähköisistä ohjeistuksista ja nettitutoriaaleista.

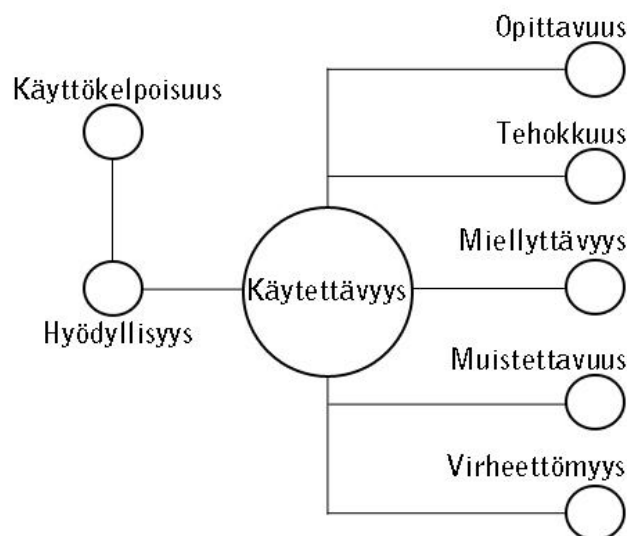
2 OHJELMAN KÄYTETTÄVYYDESTÄ

”Käytettävyys (engl. usability) on apuvälineen tai muun valmistetun esineen, palvelun tai ympäristön helppokäyttöisyyttä tietyn tavoitteen saavuttamiseksi. Käytettävyydellä voidaan myös viitata helppokäyttöisyyttä mittaaviin menetelmiin sekä oppiin niistä periaatteista joita soveltamalla tuotteesta, palvelusta tai ympäristöstä saadaan helppokäyttöisempi.

Ihmisen ja koneen välisessä vuorovaikutuksessa käytettävyydellä viitataan yleensä tietokoneohjelmiston tai verkkosivun helppokäyttöisyyteen.”

– Wikipedia1

Jakob Nielsen (1993) määrittelee käytettävyyttä monipuolisesti ohjelman tai verkkopalvelun laatuna (ks. Kuva 2).



Kuva 2 Käytettävyyden osatekijöitä Nielsenin (1993) mukaan.

Nielsenin (1993) mukaan internetpohjaisen sovelluksen käytettävyydessä tulee ottaa huomioon sovelluksen käyttötarkoitus. Käyttötarkoituksen ja siihen vaadittavien toimintojen kartoittaminen mahdollistaa hyvän käytettävyyden. Tiettyyn tarkoitukseen luodun sovelluksen tai palvelun tulee olla **käyttökelpoinen**, jotta sen käyttö nähdään **hyödylliseksi** (Kuva 2), tässä tapauksessa mainostoimistossa osana sen tarjoamia palveluja. Käyttökelpoisella sovelluksella tulee voida suorittaa siltä vaaditut toiminnot. Käyttökelpoisuuden kohdalla tulee ottaa huomioon myös ihmisten rajoitteet, kuten värisokeus ja muut ongelmat, jotka saattavat haitata verkkopalvelun käyttöliittymän ja toiminnan hahmottamista.

Nettitutoriaali-ohjelman administraattori-käyttäjiä eli sisällöntuottajia ovat mainostoimistoalan ammattilaiset, joita varten tässä opinnäytetyössä varsinaisesti ohjelma on toteutettu. Suunnittelussa ei voi unohtaa ohjelman käyttökelpoisuutta päätekäyttäjienkään kohdalla. He ovat mainostoimiston julkaiseman tutoriaalin asiakkaat, yritykset ja niiden työntekijät, jotka opastavat omia asiakkaitaan nettitutoriaalin avulla.

Mainostoimiston tulee voida tarjota tutoriaaleja verkkopalveluna mahdollisimman monille yrityksille, jotka voivat olla täysin eri alan toimijoita. Käyttökelpoisuus ja hyödyllisyys tässä tapauksessa tarkoittavat ohjelman ja palvelun linkittymistä yrityksen markkinointitavoitteisiin tarjota mainostoimiston esite- ja tutoriaalipalveluja verkkoympäristössä. Luovan ja varsinkin visuaalisen alan yritykselle tärkeää on silloin nettitutoriaalin tekninen ja visuaalinen käyttökelpoisuus.

Muita internetpohjaisten sovellusten käytettävyyden pääpilareita ovat muun muassa sovelluksen **miellyttävyys** ja **tehokkuus** (Kuva 2). Käyttäjän on voitava käyttää sovellusta ongelmitta ja koettava sen käyttö helpoksi ja johdonmukaiseksi. Sovelluksen tehokkaalla käytöllä käyttäjältä säästyy aikaa ja resursseja haluttujen toimenpiteiden suorittamiseen. Miellyttävä ja helppokäyttöinen käyttöliittymä edesauttavat sovelluksen tehokasta käyttöä. Sovelluksen käyttöliittymän kohdalla miellyttävyyteen liittyy monia seikkoja, kuten esimerkiksi hillitty värimaailma ja toimintojen looginen asemointi.

Internetpohjaisen sovelluksen tapauksessa sen toimintatehokkuuteen vaikuttavat myös tekniset seikat, kuten sovelluksen optimointi ja palvelin, jolla sivusto sijaitsee. Sovelluksen optimoinnilla taataan sen jouheva toiminta kaikissa tilanteissa. Optimointi voidaan suorittaa esimerkiksi koodin siistimisellä ja järjeistämällä, jolloin koodi suoritetaan käytön kannalta mahdollisimman loogisessa järjestyksessä. Hyvin optimoitu sovellus aiheuttaa myös vähemmän rasitusta palvelimen päässä, joka nopeuttaa sovelluksen toimintaa entisestään.

Käytettävyyden, erityisesti sen hyödynnettävyyden kannalta olennaista on myös ohjelman **virheettömyys** ja loogisuus (Kuva 2). Käyttäjän tulee voida tehdä ohjelmalla asioita, joita ohjelman ulkoasu antaa ymmärtää. Ohjelman looginen ja virheetön toiminta yhdistettynä miellyttävään ja helppokäyttöiseen käyttöliittymään antaa käyttäjälle positiivisen kuvan, tässä tapauksessa yrityksestä ja yritykselle positiivisen kuvan tarjoavasta mainostoimistosta.

Sovelluksen selkeä ja minimalistinen käyttöliittymä auttaa ensikertalaista käyttäjää oppimaan sovelluksen toiminnot helposti ja nopeasti. Käyttöliittymän nopea **opittavuus** auttaa myös sovelluksen myöhempää käyttöä, jolloin käyttäjän on helppo **muistaa** sovelluksen toiminnot (Kuva 2).

Nettitutoriaaliohjelmaan pyrittiin toteuttamaan selkeä ja helppokäyttöinen ulkoasu, joka tarjoaisi käyttäjälle nettitutoriaalien luomiseen tarvittavat työkalut ja toiminnot. Nettitutoriaaliohjelman perustyökaluilla voidaan luoda selkeitä ja minimalistisia tutoriaalisivustoja, teksti- kuva- ja videolaatikoin. Edistyneen käyttäjän ei tarvitse kuitenkaan rajoittua sovelluksen tarjoamiin perustyökaluihin, sillä tekstityökalulla voi lisätä html-koodia tutoriaalisivustolle. Mainostoimistoalan ammattilaisten voidaan olettaa olevan enimmäkseen tällaisia edistyneitä käyttäjiä.

Valmiin tutoriaalisivuston alalaidassa on helppokäyttöinen navigaatio, jolla tutoriaalin lukija voi selata tutoriaalin sivuja.

Hyvän käytettävyyden omaava ohjelma voi säästää sitä käyttävältä yritykseltä resursseja, koska yrityksen ei tarvitse tuhata aikaa ohjelman ongelmien ja epäloogisuuksien selvittämiseen. Tehokkaasti käytettävällä ohjelmalla käyttäjiltä ei myöskään kulu turhaa aikaa ohjelmalta vaadittujen toimenpiteiden suorittamiseen.

Internetpohjaisen sovelluksen tehokkaaseen ja vaivattomaan käyttöön liittyy käyttöliittymän selkeä ulkoasu, joka käyttäjän on helppo oppia ja sisäistää. Nettitutoriaaliohjelman suunnittelussa ja ohjelmoinnissa juuri tämä pyrittiin ottamaan huomioon erityisesti mainostoimiston ammattilaisten näkökulmasta, mutta myös päätekäyttäjien näkökulmasta.

*”Tyytyväinen käyttäjä on tyytyväinen asiakas.
Tyytyväinen asiakas on asiakas jatkossakin.”*

- (IHTE) Kaper s2007 Luento 6

3 NETTITUTORIAALIOHJELMAN SUUNNITTELU

Nettitutoriaaliohjelman suunnittelu ja toteutus tapahtuivat paljolti samaan aikaan erilaisten kokeilujen ja testauksen avulla. Tätä työtapaa voidaan sanoa tekniseksi suunnitteluksi (engineering design):

"...Among the fundamental elements of the design process are the establishment of objectives and criteria, synthesis, analysis, construction, testing and evaluation."

- Wikipedia 2

Wikipedia-lainauksen mukaan siis tuotteen tai kuten tässä sovelluksen teknisen suunnittelun peruselementtejä ovat ensin kriteerit, synteesi ja analyysi sekä itse työ eli sovelluksen rakentaminen, testaus ja arviointi siitä mitä pitäisi tehdä toisin ja kehittää. Kriteereitä ovat tässä Nielsenin (1993) käytettävyyuskriteerit ja se mitä olen oppinut mainostoimistoympäristöstä työharjoittelussa.

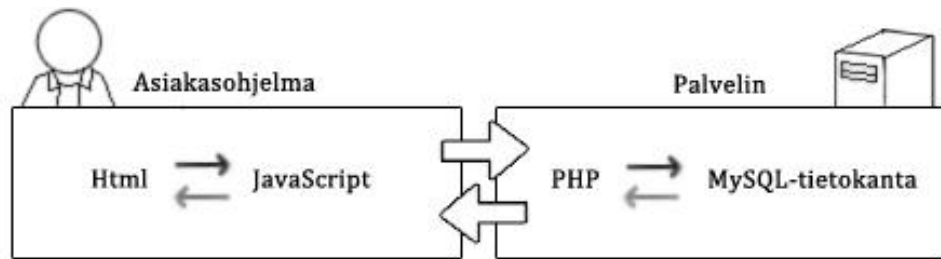
Jatkossa kuvaan sovelluksen teknistä suunnittelua ja toteutusta. Testaus: Nettitutoriaaliohjelman testaus eteni progressiivisesti ohjelman kehityksen myötä. Tarkempaa arviointia teen pohdintaluvussa 6.

3.1 Sovelluksen toimintaympäristö

Kuten mikä tahansa web-pohjainen sovellus, tarvitsee myös nettitutoriaali-ohjelma web-palvelimen, joka tukee sovelluksen luonnissa käytettyjä web-ohjelmointikieliä. Tässä tapauksessa web-palvelimelta vaaditaan php ja MySQL-tuki. Php on yleisin dynaamisten web-sivujen luonnissa käytetty ohjelmointikieli, joten sopivan web-palvelimen löytäminen tai luonti ei ole vaikeaa. Nettitutoriaaliohjelman käyttäjä tarvitsee modernin internetselaimen, kuten uusimman version FireFox tai Google Chrome – selaimesta.

3.2 Sovellushierarkia ja toiminta

Nettitutoriaaliohjelma sijaitsee web-palvelimella ”tutmaker” nimisessä kansiossa. Kansion juuressa on sovelluksen tarvitsemat html-, php-, css-JavaScript-tiedostot, sekä kolme kansiota, joita ovat: images, jscolor ja tutorials. Images kansiossa on nettitutoriaaliohjelman käyttöliittymässä käytettävät kuvat. Jscolor kansio taas pitää sisällään ulkoisen JavaScript-työkalun, jota käytetään tutoriaalivärien valitsemiseen. Tutorials kansioon sijoitetaan kaikki käyttäjien luomat tutoriaalit niiden omilla nimillään. Tutoriaalikansiot pitävät sisällään kyseisen tutoriaalivärien css-tyylitiedoston, tutoriaalivärien html-tiedostot, sekä navigoinnin html-tiedoston. Tutoriaalivärien kansiossa on myös kaksi alikansiota, joita ovat images ja videos. Kyseiset kansiot on varattu käyttäjän omille tiedostoille, joita voidaan lisätä tutoriaaliin.



Kuva 3 Asiakasohjelman ja palvelimen päässä tapahtuvat toimenpiteet

Käyttäjän html-tekstikenttiin syöttämä tieto lähetetään JavaScript-funktion avulla php-tiedostolle, jossa suoritetaan toimintokohtaiset MySQL tietokantakomennot. Tietokannasta ladattaessa suoritetaan php-tiedostossa funktio, joka hakee tietokannasta halutun informaation. Ladattu informaatio käsitellään edelleen JavaScript-funktion avulla (ks. Kuva 3).

3.2.1 Valmiin sovelluksen ominaisuudet

Valmiilla sovelluksella tulee voida luoda havainnoillistavia ja helposti ymmärrettäviä internetsivuja, joilla neuvotaan tietyn tuotteen tai asian käyttö. Ohjeistus- eli tutoriaalisivustolle täytyy voida tarvittaessa lisätä uusia sivuja ja hallinnoida sivukohtaista sisältöä. Käyttäjän luoma ohjeistussivusto voi koostua muunmuassa teksti- ja kuvalaatikoista. Sovelluksen tulee tarjota lisämahdollisuuksia edistyneille käyttäjille. Lisämahdollisuuksiin kuuluu esimerkiksi html-koodin syöttäminen tekstilaatikkoon, joka mahdollistaa sovelluksen täyden käytettävyyden. Syöttämällä html-koodia käyttäjä voi muokata sivun ulkoasua esimerkiksi luomalla taulukoita ja listoja.

Koska sovellus tulee mainostoimistokäyttöön, täytyy sillä voida muokata tutoriaalisivuston tyyliasetuksia. Tyyliasetuksiin kuuluvat muun muassa taustavärit ja taustakuvat. Värienhallinta on tärkeätä, koska ohjeistusta haluava yritys saattaa haluta sivustolle yrityksensä tunnusvärit.

3.2.2 MySql-tietokannan rakenne

Jokaiselle käyttäjän luomalle tutoriaalille luodaan oma tietokanta, johon tallennetaan tutoriaalın tyyliasetukset, sivut ja sivujen elementit. Tutoriaalın sivuille luodaan omat taulut, jotka nimetään niiden järjestysnumeron mukaan seuraavasti: page1, page2,... Sivutaulu pitää sisällään vain sivun otsikon, joka kirjoitetaan tietokantaan "<div>" html-komennon sisään.

Sivuille lisättävien teksti-, kuva- ja video-elementtien tiedot tallennetaan myös omiin tauluihinsa, jotka nimetään numerolla, joka määrää mille sivulle elementti kuuluu. Elementeille annetaan myös omat järjestysnumerot. Elementtitaulut nimetään siis seuraavasti: element1_1, element1_2,... Elementtitaulu pitää sisällään elementin tyyppin, alatyypin, elementin toiminnan kannalta oleelliset html-komennot, sekä otsikon ja tekstin.

3.3 Toteutuksessa tarvittavat tekniikat

Alkuperäinen suunnitelma oli luoda nettitutoriaaliohjelmasta työpöytäsovellus Java-ohjelmointikielellä, mutta päädyin lopulta selainpohjaiseen ratkaisuun, koska selainpohjainen sovellus toimii lähes missä vain.

Selainpohjaista sovellusta suunnitellessa tulee eteen monia toteutusvaihtoehtoja asiakas- ja palvelinpään ohjelmointiin. Asiakaspään ohjelmointi voidaan toteuttaa muun muassa JavaScript, Flash tai Microsoft Silverlight –skriptikielellä. Päädyin JavaScript-kieleen, koska se on toimintavarmin. Valmiin nettitutoriaalisovelluksen toimintaympäristönä tulee olemaan mainostoimiston tekninen ympäristö, jossa ovat käytössä Mac-tietokoneet. Sovelluksen visuaalisen ilmeen ja päärunгон toteutukseen käytin normaaleja web-sivujen luontiin tarkoitettuja tekniikoita, kuten html-kieltä ja css-tyylejä.

Palvelinpään ohjelmointiin löytyy myös monia eri vaihtoehtoja, joita ovat muunmuassa ASP, Java EE, Perl, Python, Ruby ja Php. Päädyin käyttämään php-kieltä, koska se on vaihtoehtoista ainoa, josta minulla oli aiempaa kokemusta.

3.3.1 Html, Css, JavaScript

Html (Hypertext markup language) on kieli, jota käytetään internetsivujen luomiseen. Html-koodilla ei voida kuitenkaan luoda varsinaista toiminnallisuutta sivuille. Html-kielellä voidaan luoda internetsivulle runko, painikkeet ja tekstilaatikot. Nettitutoriaaliohjelman luonnissa on käytetty html-kielen versiota 4.01, joka on vielä laajalti käytössä, vaikka html-kielen uusin versio on 5.0.

Css (Cascading style sheets) on html-kielen ohessa käytettävä tyylikieli, jolla voidaan määritellä html-elementtien tyyliasetuksia. Css-tyylikielellä voidaan määritellä esimerkiksi tiettyjen html-elementtien taustavärit, fonttikoot ja paljon muuta.

JavaScript on pääasiassa web-ympäristöön tarkoitettu scriptikieli, jolla voidaan luoda toiminnallisuutta ja dynaamisia elementtejä html-sivuille. Toiminnallisuus rajoittuu kuitenkin vain asiakasohjelman, eli tässä tapauksessa internetselaimen päähän. Tämä tarkoittaa sitä, että JavaScript –scriptikielellä ei voida hallinnoida asiakasohjelman ulkopuolisia asioita, kuten esimerkiksi mysql-tietokantoja.

3.3.2 Php, MySql

Php (PHP: Hypertext Preprocessor) on ohjelmointikieli, jota käytetään web-palvelinympäristöissä. Php-koodi suoritetaan web-palvelimella ja sillä voidaan hallinnoida tietokantoja, kuten MySql-tietokannat.

MySQL on relaatiotietokantaohjelmisto, jonka päälle tehtävä ohjelmalogiikka tehdään usein php, perl tai python-ohjelmointikielellä. Tietokantoja käytetään tiedon varastointiin, kuten esimerkiksi tässä tapauksessa tutoriaalivivut ja niiden sisällä olevat sisältöelementit.

3.4 Käyttöliittymän suunnittelu

Käytin käyttöliittymän suunnittelussa apuna Jakob Nielsenin (1993) luomia heuristiikkoja. Nielsenin heuristiikat koostuvat kymmenestä säännöstä, joita noudattamalla voidaan luoda käyttäjäystävällisiä käyttöliittymiä. Nielsenin heuristiikoista pidin erityisesti silmällä seuraavaa viittä sääntöä:

- Yhteneväisyys ja standardit
- Virheiden estäminen
- Tunnistaminen mieluummin kuin muistaminen
- Käytön joustavuus ja tehokkuus
- Esteettinen ja minimalistinen design

Jakob Nielsenin mukaan hyvässä käyttöliittymässä on yhteneviä elementtejä, jotka käyttäjän on helppo hahmottaa ja tunnistaa toimintojen käyttötarkoitus ilman erillistä kokeilua. Toistuvat elementit auttavat myös käyttäjää muistamaan painikkeiden ja työkalujen toiminnot.

Virhetilanteiden estäminen on tärkeä ominaisuus hyvässä käyttöliittymässä. Sovelluksen ja käyttöliittymän tulisi toimia siten, että käyttäjän on mahdollonta saada aikaan virhetilanne. Jos virhetilanne kuitenkin ilmaantuu, tulee siitä ilmoittaa käyttäjälle selkeästi.

Tältä pohjalta suunnittelin sovelluksen graafisen käyttöliittymän mahdollisimman yksinkertaiseksi ja helposti omaksuttavaksi. Graafinen käyttöliittymä koostuu esikatseluikkunasta ja ruudun oikeassa reunassa sijaitsevasta työkalupalkista (ks. Kuva 4).



Työkalupalkin toiminnot on järjestetty tutoriaalin luonnin kannalta loogiseen järjestykseen. Kaikki työkalut sijaitsevat ryhmitettyinä omiin työkalulaatikoihinsa. Käyttöliittymää tehdessäni huomasin eri selainten ja käyttöjärjestelmien vaikutuksen html-painikkeisiin. Macintoshin käyttöjärjestelmässä painikkeet saivat pyöristetyn muodon ja olivat hieman erikokoisia kuin Windows-käyttöjärjestelmässä. Vastaavanlaisia ongelmia ilmeni myös eri selainten tavassa piirtää painikkeita. Ratkaisin ongelman luomalla painikkeille yhteisen ulkoasun. Suunnittelin uuden ulkoasun, niin että painikkeilla on omat värinsä. Lisäyspainikkeet ovat vihreitä, muuttamispainikkeet keltaisia ja poistamispainikkeet punaisia. Eriväristen painikkeiden myötä käyttäjän on helppo huomata tiettyjen painikkeiden toistuvuus ja siten ymmärtää niiden toiminta.

Ensimmäisessä työkalulaatikossa on tutoriaalin hallintaan liittyvät työkalut, kuten uuden tutoriaalin luonti, avaus ja poisto. Samassa laatikossa on myös tutoriaalin sivujen hallintatyökalut, kuten sivun luonti, otsikon muuttaminen ja sivun poisto, sekä esikatseluikkunan päivittäminen. Tutoriaalin ja sivujen hallinnan vieressä on tyhjä tila, johon listautuvat luodut sivut. Luodut sivut listataan ylhäältä alaspäin painikkeina, joissa lukee kyseisen sivun numero. Seuraavaksi käyttöliittymässä on alasvetovalikko, johon listataan tutoriaaliin lisätyt elementit, kuten tekstit, kuvat ja videot. Alasvetovalikosta käyttäjän on helppo valita haluamansa elementti. Valitun elementin tiedot ladataan elementinlisäystyökalun tekstikenttiin.

Elementtien lisäykseen käytettävässä työkalulaatikossa on kolme välilehteä. Jokaisella välilehdellä on toistuvana elementtinä painikkeet, joilla voi lisätä, muuttaa tai poistaa kyseisen elementin. Ensimmäinen välilehti pitää sisällään tekstin lisäykseen käytettävät työkalut. Tekstiä lisätessään käyttäjä voi valita tekstilaatikon otsikolle haluamansa otsikkotyylin. Otsikkotyylin vaihto käy helposti radio-painikkeiden avulla. Seuraavalla välilehdellä on kuvaelementtien lisäys. Kuvaa lisätessä käyttäjä voi valita radio-painikkeiden avulla kuvatekstin sijainnin. Radio-painikkeiden alla on kaksi tekstikenttää, joista ensimmäiseen syötetään kuvan polku ja toiseen kuvateksti. Kolmannella välilehdellä on videonlisäykseen tarvittavat työkalut. Lisättävän videon formaattivaihtoehtoina ovat Youtube, wmv ja .mov. Valinta tapahtuu radio-painikkeilla. Painikkeiden alla on tekstikenttä johon syötetään videon polku. Youtube-videon tapauksessa kenttään tulee syöttää videon koodi, joka löytyy videon Url-osoitteesta.

Viimeinen työkalulaatikko on tutoriaalin tyylin hallinta. Tyyliasetuksiin kuuluu mm. Taustaväri, taustakuva, sivuleveys ja fonttikoko. Värien valitseminen hoituu helposti Jscolor-väriavalitsimella.

Jokainen työkalulaatikko sisältää ohjeistuspainikkeen, jota painamalla käyttäjä voi avata ohjeistussivuston, joka aukeaa uuteen selainikkunaan. Ohjeistussivusto on tehty tietenkin samaisella nettitutoriaaliohjelmalla (ks. Kuva 4).

Tutoriaalinhallinta

Uuden tutoriaalin lisääminen

Kirjoita avattavan tutoriaalin nimi tekstikenttään ja paina Luo uusi -painiketta. Uusi tutoriaali on onnistuneesti luotu, kun sivulistaukseen ilmestyy pääsivun painike.

Tutoriaalin avaaminen

Kirjoita avattavan tutoriaalin nimi tekstikenttään tai valitse avattava tutoriaali tekstikentän alla olevasta alasvetovalikosta. Paina lopuksi Avaa -painiketta.

Tutoriaalin poistaminen

Kirjoita poistettavan tutoriaalin nimi tekstikenttään ja paina Poista -painiketta. Tutoriaalin poistaminen poistaa tutoriaalin vain tietokannasta. Tutoriaalin tiedostojen poistaminen tulee hoitaa manuaalisesti.

Sivujenhallinta

Uuden sivun lisääminen

Voit luoda uuden sivun painamalla Luo uusi sivu -painiketta. Sivun otsikko ei ole pakollinen, mutta voit sen halutessasi lisätä painikkeiden päällä olevaan tekstikenttään.

Otsikon muuttaminen

Sivun otsikon voi muuttaa menemällä kyseiselle sivulle ja kirjoittamalla uuden otsikon painikkeiden päällä olevaan tekstikenttään. Hyväksy muutos painamalla Muuta otsikko -painiketta.

Sivun poistaminen

Avaa poistettava sivu ja paina Poista sivu -painiketta. Poistettu sivu poistetaan tietokannasta ja sen html-tiedosto poistetaan tutoriaalin kansiota. Sivun yhteydessä olleita kuvia ja video -tiedostoja ei poisteta.

1
2
3
4

Kuva 5 Nettitutoriaali ohjelmalla tehty tutoriaalisivusto

Jokainen tutoriaalisivusto koostuu keskitetystä keskikaistaleesta, jonka sisään luotava sisältö sijoitetaan. Keskikaistaleen alapuolella on navigointilinkit, joilla tutoriaalin lukija voi navigoida tutoriaalisivustolla (ks. Kuva 5).

4 NETTITUTORIAALIOHJELMAN TOTEUTUS

4.1 Toteutuksessa käytetyt sovellukset

Kaikki käyttämäni sovellukset ovat ilmaisia ja vapaasti netistä ladattavia.

Aloitin ohjelman koodaamisen PSPad nimisellä ohjelmalla. PSPad-ohjelmasta löytyy tuki monelle nettipohjaisen sovelluksen koodamiseen tarvittavalle kielelle. Eteen tuli kuitenkin ongelma JavaScript-koodin värjäyksen kanssa. PSPad ei värjännyt koodia haluamallani tavalla ja teki siitä epäselvää luettavaa. Ratkaisin ongelman lataamalla NetBeans-ohjelman, jossa on paremmat tekstinvärjäysominaisuudet.

NetBeans on ilmainen ohjelmointiympäristö, joka on tarkoitettu monille eri ohjelmointikielille. NetBeans on suunniteltu pääasiassa Java-ohjelmointikielelle, mutta ohjelmasta löytyy tuki myös php-kielelle.

Php:ta hyödyntävät sivustot tarvitsevat Php-koodia tukevan palvelimen. Käytössäni ei ollut oikeaa php-palvelinta tähän projektiin, joten latasin Xampp nimisen virtuaalipalvelinohjelman. Xampp-ohjelmalla loin virtuaalisen Php- ja MySQL-palvelimen. MySQL-tietokantojen hallintaan käytin Xampp:n mukana tullutta Phpmyadmin selainpohjaista sovellusta.

4.2 Kehityksen kulku

Suunniteltuani sovelluksen pääkäyttöliittymän graafisen ilmeen aloin tekemään siitä rautalankamallia html-kielellä. Alkuperäinen html-runko oli yksinkertainen, eikä sisältänyt varsinaisia toimintoja. Aloitin tutoriaalin luonnista. Esimerkkien avulla sain luotua JavaScript-funktion, joka lähettää tutoriaalin nimen muuttujassa ulkoiselle Php-tiedostolle. Php-tiedostossa luodaan tutoriaalin niminen tietokanta MySQL-tietokantaan.

Tässä vaiheessa olin luonut valmiiksi painikkeet muillekin ominaisuuksille, kuten tutoriaalin avaus ja poisto. Jätin niiden toiminnallisuuksien ohjelmoinnin kuitenkin myöhemmälle. Seuraavaksi halusin kokeilla html-tiedostojen luomista palvelimelle. Tiesin, että tämä tulisi olemaan tärkeä ominaisuus, koska koko ohjelman toiminnallisuus perustuisi html-sivuille, jotka ladattaisiin pääkäyttöliittymän iframe-ikkunaan. Onnistuin luomaan html-sivun tutoriaalin luonnin yhteydessä.

Onnistuneen kokeilun jälkeen loin ohjelman html-runkoon iframen johon luodut sivut avattaisiin. Seuraavaksi tein JavaScript-funktion, joka päivitti iframen avaamalla siihen juuri luodun sivun. Php-tiedostossa luodulla html-sivulla ei kuitenkaan ollut sisältöä.

Seuraava ominaisuus, jota lähdin työstämään olikin saada käyttäjän syöttämä sisältö näkymään html-sivulla. Aloitin tekemällä tekstilaatikon ja painikkeen ohjelman pääkäyttöliittymään. Tarkoituksena oli syöttää tekstiä tekstilaatikkoon ja lähettää teksti painamalla painiketta ja täten

saada teksti näkymään luotavalla html-sivulla. Tekstin lähettäminen tapahtuu samalla JavaScript-funktiolla kuin tutoriaalin luonti. Tekstilaatikon teksti lähetetään siis muuttujassa ulkoiselle php-tiedostolle, jossa teksti kirjoitetaan luotavaan html-sivuun. Tässä vaiheessa tekstiä ei vielä tallennettu tietokantaan.

Tein sivujen luonnille painikkeen, jolla käyttäjä voi lisätä uusia sivuja tutoriaaliin. Seuraavaksi ohjelmoin logiikan, jolla luodut sivut numeroidaan ja tietokannan taulut luodaan. Luodut sivut eivät tässä vaiheessa vielä näkyneet ohjelman pääkäyttöliittymässä. Alkuperäinen ratkaisuni ongelmaan oli alasvetovalikko, josta käyttäjä voisi vaihtaa sivua. Päädyin kuitenkin sivupainikkeisiin, koska painikkeet ovat visuaalisesti ja toiminnollisesti parempi vaihtoehto. Sivun luonnin yhteydessä luodaan siis sivulle oma painike, jota painamalla käyttäjä voi päivittää pääkäyttöliittymän iframe-ikkunan haluamallaan sivulla.

Toimivan sivujenluontilogiikan jälkeen palasin sisällön tuottoon. Ohjelmoin toiminnon, joka tallensi käyttäjän syöttämän tekstin tietokantaan valitun sivun tauluun. Muutin toiminnon myöhemmin sellaiseksi, että jokaiselle lisätylle sisältöelementille tehdäänkin tietokantaan oma taulu, joka nimetään kyseisen sivun ja elementin järjestysnumeron mukaan. Tiesin että kyseinen muutos helpottaisi tulevaisuudessa sivun elementtien hallinnointia liittyen elementtien poistamiseen ja laskemiseen. Loin myös toiminnon, jolla pystyy tallentamaan elementtien ja sivun otsikon muutokset tietokantaan. Muutostoimenpide ei tässä vaiheessa vielä kirjoittanut sivun html-tiedostoon.

Seuraavaksi halusin luotaville sivuille tyyliä, koska tähän mennessä ne olivat vain valkostaustaisia sivuja, joilla oli mustaa tekstiä. Muutin sivunluontikoodia siten, että sivua luodessa sinne kirjoitetaan html-kielellä oma runkonsa. Myöhemmin lisäsin luotaville sivuille myös oman .css – tyylitiedoston, joka mahdollisti sivun tyyliasetusten muokkaamisen.

Tässä vaiheessa käyttäjä pystyi siis luomaan tutoriaalin, lisäämään siihen sivuja ja sivuille tekstiä. Tutoriaalin sivuja ja tekstilaatikoita ei siis voinut vielä poistaa. Tekstilaatikoiden sisältöä ei voinut myöskään muokata.

Pohtiessani sivujen ja elementtien poiston logiikkaa aloin tekemään kuvienlisäystoimintoa. Se oli suhteellisen helppo luoda, koska sen toimintaperiaate on samanlainen kuin tekstinlisäys. Asetin kuvanlisäystyökalulaatikon sovelluksen pääkäyttöliittymässä tekstinlisäyslaatikon alle. Toimivan kuvan lisäyksen jälkeen palasin elementtien poisto-ominaisuuden suunnittelemiseen. Ensin loin alasvetovalikon, johon lisätyt elementit listataan. Sen jälkeen loin Javascript-funktion, joka ottaa alasvetovalikosta valitun kohdan numeerisena arvona ja lähettää sen edelleen Php-tiedostolle. Php-tiedostossa etsitään alasvetovalikosta valitun numeron perusteella oikea taulu. Valitun taulun tiedot avataan tekstinlisäystyökalun tekstikenttiin. Seuraavaksi loin poistotoiminnon, joka poisti käyttäjän valitseman elementin. Tässä vaiheessa poistotoiminto yksinkertaisesti vain poisti

valitun elementin taulun tietokannasta. Tämä johti siihen, että elementtien numerointi ei enää pätenyt poistetun elementin jälkeen. Korjasin asian tekemällä silmukkarakenteen, joka nimeää taulut uudestaan poiston jälkeen. Tein myös Javascript-funktion, joka päivittää alavetovalikon elementin poistamisen jälkeen. Sovelsin toimivaa elementin poistologiikkaa sivun poistoon onnistuneesti. Sivun poiston yhteyteen loin JavaScript-funktion, joka päivittää sivupainikelistan sivun poiston yhteydessä. Sivun poisto poistaa myös kaikki sivun elementit ja tuhoaa sivun html-tiedoston.

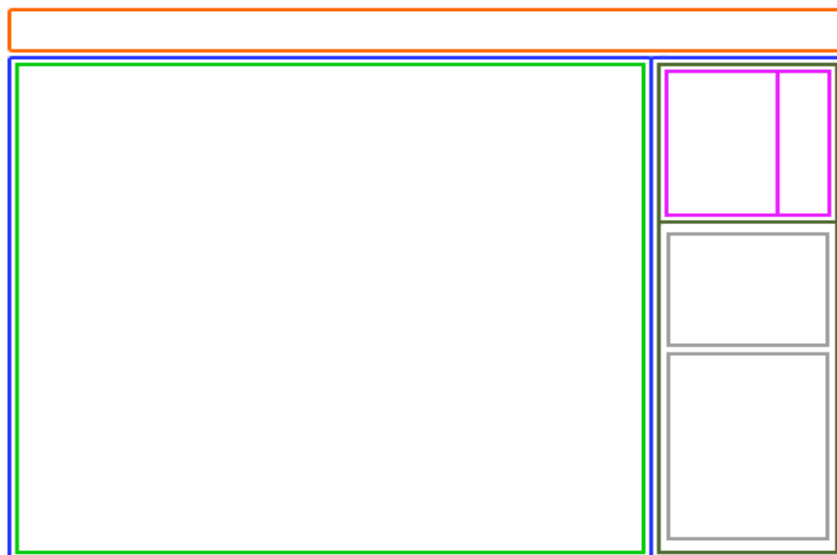
Seuraavaksi loin ulkoisen Php-luokan, joka kirjoittaa Html-tiedostoon. Luokkaa käytetään aina kun elementti lisätään, poistetaan tai kun elementin sisältöä muokataan.

Tämän jälkeen loin kuvanlisäykselle erilaiset muotoiluvaihtoehdot. Erilaisia muotoiluvaihtoehtoja on kolme erilaista: pelkkä kuva, kuva ja tekstilaatikko kuvan vasemmalla puolella, sekä kuva ja tekstilaatikko kuvan oikealla puolella. Tein myös työkalulaatikon videoiden lisäykselle. Sovelluksen käyttöliittymässä oli siis nyt neljä työkalulaatikkoa allekkain, joka johti siihen, että käyttäjä joutui rullaamaan sivua alaspäin nähdäkseen videon lisäyksen työkalut. Korjasin ongelman luomalla kätevän välilehtisysteemin, jonka avulla sain työkalut samaan laatikkoon. Toimivien sisällönlisäystyökalujen jälkeen tein alavetovalikon, johon lisäävät sisältöelementit listataan. Alavetovalikosta käyttäjä voi valita aiemmin lisätyn elementin muokattavaksi.

Lopuksi tein alavetovalikon, josta käyttäjä voi kätevästi valita avattavan tutoriaalin. Kyseiseen alavetovalikkoon listataan kaikki palvelimen tietokannat. Tein myös työkalulaatikkokohtaiset info-painikkeet, joita painamalla käyttäjä voi avata ohjeistussivuston.

4.2.1 Html-runko

TutMaker sovelluksen pääsivu on palvelimella Tutmaker-kansiossa nimellä index.html. Sovelluksessa tarvittavat JavaScript- ja Php-koodit ladataan ulkoisista tiedostoista, jotka sijaitsevat palvelimella Tutmaker-kansion juuressa. Poikkeuksena jscolor.js JavaScript-tiedosto, joka on Tutmaker- kansion alikansiossa nimeltä Jscolor.



Kuva 6 Nettitutoriaaliohjelman html-runko

Ohjelman html-runko koostuu kuvan mukaisista taulurakenteista (ks. Kuva 6). Oheisen kuvan värien mukaisesti oranssi taulu on varsinaisen käyttöliittymätaulun ulkopuolella. Oranssin taulun tarkoitus on olla visuaalinen elementti. Kuvan sininen taulu on ohjelman päätaulu, jonka sisällä muut taulut ovat. Sininen taulu on jaettu kahteen soluun, joista vasemmanpuoleinen toimii eräänlaisena esikatseluikkunana ja oikeanpuoleinen pitää sisällään tutoriaalin hallinnassa tarvittavat työkalut. Sinisen taulun vasemmassa solussa on taulu, jonka sisällä on iframe-ikkuna. Iframe-ikkunaa tarvitaan tutoriaalisivujen näyttämiseen käyttöliittymässä. Sinisen taulun oikeanpuoleisessa solussa on taulu, joka on kuvassa merkitty tumman vihreällä. Tumman vihreä taulu on jaettu pystysuunnassa kahteen soluun. Ylemmässä solussa on violetti taulu, jonka sisällä on mm. tutoriaalin ja tutoriaalin sivujen hallintaan tarvittavat toiminnot. Tumman vihreän taulun alemmassa solussa on muut tutoriaalin hallinnoinnissa käytettävät työkalut kuten tekstilaatikoiden ja kuvien lisäys sekä tyylitiedoston muokkaus. Kyseiset työkalut ovat omissa div-laatikoissaan, jotka näkyvät kuvassa harmaana.

4.2.2 JavaScript ja Php –toiminnot

Sovelluksessa käytettävät JavaScript-funktiot ovat ulkoisessa javascript-tiedostossa, jonka nimi on tutmakerjs.js. JavaScriptillä hoidetaan käyttäjän päässä suoritettavat toimenpiteet, kuten painikkeiden toiminnot, iframe-esikatseluikkunan päivitys ja muut käyttöliittymän toiminnot. Toinen sovelluksessa käytettävä javascript-tiedosto on jscolor.js, joka sijaitsee omassa kansiossaan nimeltä jscolor. Jscolor on ilmainen javascriptillä toteutettu värinvalitsin (<http://jscolor.com/>), jota käytetään sovelluksen tyylitiedoston väriattribuuttien muuttamiseen.

Sovelluksessa käytettävät php-toiminnot on pilkottu moneen eri tiedostoon, jotka suoritetaan käytettävän työkalun vaatiessa. Php-koodilla hoidetaan palvelimen päässä tapahtuvat asiat, kuten tietokantaan tallentaminen ja tietokannasta lataaminen. Tietokannanhallinnan lisäksi

php:llä hoidetaan myös tiedostojen luonti palvelimelle ja tiedostojen muokkaus.

- tutorial_ctrl.php
Suorittaa tutoriaalin luonnin, avaamisen ja poiston. Uuden tutoriaalin kohdalla suoritettaviin toimintoihin kuuluu mm. tietokannan, tutoriaalin kansiorakenteen, pääsivun ja css-tiedoston luonti. Avaamisen yhteydessä luetaan tietokannasta halutun tutoriaalin sivujen ja elementtien määrä, sekä tutoriaalin tyyliasetukset. Tutoriaalin poisto poistaa halutun tutoriaalin tietokannan. Tutoriaalin kansiorakenteen käyttäjä joutuu poistamaan manuaalisesti.
- page_ctrl.php
Tutoriaalin sivujen toiminnot, kuten sivun luonti, otsikon muuttaminen ja sivun poisto. Sivun lisäys luo sivulle uuden taulun tietokantaan, sekä sivun html-tiedoston. Sivun lisäyksen yhteydessä muokataan myös tutoriaalisivujen alalaidassa olevaa navigointipalkkia.
- element_ctrl.php
Tekstilaatikoiden, kuvien ja videoiden lisäys, muuttaminen ja poisto. Jokaisessa tapauksessa luodaan tietokantaan elementtitaulu, joka pitää sisällään elementin tyyppin, otsikon, tekstin ja muita tyyppikohtaisia tietoja. Jokaisen toiminnon lopussa suoritetaan clsDrawPage.php-luokka.
- clsDrawPage.php
Ulkoinen php-luokka, jota käytetään monissa eri tilanteissa, kuten uuden sisältöelementin luonnissa, muuttamisessa ja poistossa. Luokka lataa tietokannasta sivun tiedot ja kirjoittaa ne sivun html-tiedostoon. Luokka palauttaa elementtien lukumäärän, joka tallennetaan edelleen muuttujaan. Kyseistä Muuttujaa käytetään esimerkiksi alavetovalikon täyttämiseen.
- style_ctrl.php
Tyyli-tiedoston muuttaminen. Tyylien attribuutit tallennetaan tietokantaan ja kirjoitetaan tutoriaalin css-tyyli-tiedostoon.
- change_textbox.php
Suoritetaan, kun käyttäjä valitsee alavetovalikosta haluamansa sisältöelementin, kuten tekstilaatikon. Hakee tietokannasta valitun elementin tekstisisällön.
- get_textboxes.php
Laskee ladatun sivun sisältöelementit. Elementtien määrää käytetään alavetovalikon täyttämiseen oikealla määrällä kohteita.

Php-tiedostot liitetään osaksi sivuston toiminnallisuutta JavaScript-funktiolla, joka välittää tarvittavat muuttujat php-tiedostoille.

4.3 Sovelluksen testaus

Sovelluksen toimintaa testatessani käytin pääasiassa Xampp-virtuaalipalvelinta ja Google Chrome –internetselainta Windows-ympäristössä. Työpäivinä Preciksessä käytin Xampp-virtuaalipalvelinta ja Mozilla Firefox -internetselainta Mac-ympäristössä. Xampp'in toiminta ei eronnut eri käyttöjärjestelmillä, mutta internetselainten välinen ero oli huomattava. Ongelmia tuli esimerkiksi sovelluksen pääkäyttöliittymän työkalupaneelin kanssa, koska selaimet tulkitsevat Css-tyylitiedostoja hieman eri tavalla. Ongelmat olivat siis pääasiassa visuaalisia. Eroavaisuuksia löytyi kuitenkin myös toiminnallisuuden puolelta, sillä jotkin Javascript-funktiot toimivat eri tavalla selaimesta riippuen. Toiminnallisuusongelmia ilmeni esimerkiksi iframe-ikkunan päivityksessä.

Sain korjattua pahimmat ongelmat, mutta Firefox -selaimella ilmenee vielä satunnaisia iframen automaattiseen päivitykseen liittyviä ongelmia. Ratkaisin ongelman osittain luomalla painikkeen, jolla käyttäjä voi päivittää iframe-ikkunan. Epäilen, että ongelma johtuu selaimen tavasta käsitellä sivuja, jotka ovat välimuistissa. Selain avaa siis vanhan sivun välimuistista, vaikka oikea tiedosto on päivittynyt.

Testasin sovellusta muutaman kerran myös Safari- ja Internet Explorer –internetselaimilla. Safarilla sovellus toimi muutamia visuaalisia virheitä luukuunottamatta hyvin, mutta Internet Explorerilla ilmeni suurempia ongelmia. Internet Explorerin ongelmat olivat myös lähinnä visuaalisia, mutta luonteeltaan sellaisia, että ne rikkoivat koko käyttöliittymän. Päätin jättää Internet Explorerin ongelmat rauhaan, koska Preciksessä käytetään vain Firefoxia ja Safaria.

5 NETTITUTORIAALIOHJELMAN OMINAISUUDET

Tutoriaalisovelluksen perusominaisuuksiin kuuluvat muun muassa tekstin ja kuvien lisääminen. Tutoriaaleihin voidaan lisätä myös Youtube, Windows Media Video (wmv) ja Apple Quick Time –videoita. Ohjelmalla voi muokata myös tutoriaalisivuston tyylitiedostoa. Tyylitiedostoa muokkaamalla voi esimerkiksi vaihtaa tutoriaalisivuston värejä tai tekstilaatikoiden fonttia. Ohjelman käyttöliittymä koostuu helposti ymmärrettävistä hallintalaatikoista, joissa on värjätyt painikkeet. Painikkeet ja niiden värit toistuvat jokaisessa hallintalaatikossa.

Kaikki nettitutoriaaliohjelman työkalut suorittavat omalle toiminnallisuudelleen omistetun JavaScript-funktion, jonka sisällä suoritetaan Attach_file –funktio. Attach_file-funktiolla liitetään kyseisen työkalun Php-tiedosto osaksi sivuston toiminnallisuutta.

```
function attach_file( p_script_url )
{
    script = document.createElement( 'script' );
    script.src = p_script_url;
    document.getElementsByTagName( 'head' )[0].appendChild(
script );
}
```

5.1 Tutoriaalin hallinta



Kuva 7 Tutoriaalin hallinta

Kaikki Tutoriaalin hallinta-osion painikkeet suorittavat tutorial_ctrl Javascript-funktion, joka ottaa sisäänsä painetun painikkeen id:n ja lähettää sen edelleen attach_file - funktiolla tutorial_ctrl.php:lle. Tutorial_ctrl.php tiedostossa jatkotoimenpiteet määräytyvät painetun painikkeen id:n mukaan. Painikkeiden id:tä ovat: tut_new, tut_open ja tut_del eli uuden tutoriaalin luonti, tutoriaalin avaus ja tutoriaalin poisto.

Tutorial_ctrl.php tiedostossa painikkeiden toiminnot erotellaan if/else if -ehtolausein.

```
if ($sender == "tut_new"), else if ($sender == "tut_open")
ja else if ($sender == "tut_del")
```

Tutorial_ctrl.php tiedoston php-skriptin jälkeen suoritetaan tiedoston lopussa olevat kaksi javascript-komentoa. Toinen Javascript-komento suorittaa refreshIframe-funktion, joka päivittää käyttöliittymän iframe-ikkunan ja avaa siihen tutoriaalin pääsivun. Funktio ottaa sisäänsä path-

muuttujan, joka on tutoriaalin polku palvelimella.

```
function refreshIframe(path)
{
    document.getElementById("content").src = path;
}
```

Toinen komento suorittaa *Opentut*-funktion, jolle lähetetään mm. avattavan tutoriaalin sivumäärä, elementtimäärä ja tyyliasetukset. Funktion sisällä tarkistetaan painettu painike if/else if -lausein samaan tapaan kuin tutorial_ctrl.php-tiedostossa.

5.1.1 Tutoriaalin luonti

Luodakseen uuden tutoriaalin tulee käyttäjän syöttää tekstilaatikkoon tutoriaalille haluttu nimi ja sen jälkeen painaa Luo uusi -painiketta (ks. Kuva 7).

Painikkeen painaminen suorittaa tutorial_ctrl Javascript-funktion, joka lähettää tutoriaalin nimen eteenpäin tutorial_ctrl.php tiedostolle. Tutorial_ctrl Php-tiedosto luo uuden MySQL-tietokannan käyttäjän syöttämällä nimellä. Php-tiedosto luo palvelimelle myös tutoriaalin kansiorakenteen ja css-tyylitiedoston. Uuden tutoriaalin luonti luo automaattisesti tutoriaalille indeksisivun, joka toimii tutoriaalin pääsivuna. Uudelle tutoriaalille luodaan myös navigointisivu, joka näkyy jokaisella tutoriaalin sivulla olevassa iframe-ikkunassa.

Uuden tietokannan luonti tapahtuu normaaleilla MySQL-komennoilla. Seuraavilla komennoilla luodaan käyttäjän syöttämällä nimellä uusi tietokanta ja valitaan se käyttöön:

```
mysql_query("CREATE DATABASE " . $tutname, $connection) or
die(mysql_error());
mysql_select_db($tutname, $connection) or
die(mysql_error());
```

Seuraavaksi tutoriaalin tietokantaan luodaan taulu tutoriaalin tyyille. Tyylitaulu koostuu mm. tutoriaalisivun taustaväristä, taustakuvasta, reunuksien ja tekstien asetuksista.

```
mysql_query("CREATE TABLE style(bgcolor_a VARCHAR(50),
bgcolor_b VARCHAR(50), bgimg_a VARCHAR(500), bgimg_b
VARCHAR(500), sitewidth VARCHAR(50), brdrwidth VARCHAR(50),
brdrcolor VARCHAR(50), brdroption VARCHAR(50) , fontoption
VARCHAR(50), fontcolor VARCHAR(50), fontsize VARCHAR(50),
textboxbg VARCHAR(50), headoption VARCHAR(50), headcolor
VARCHAR(50), headsize VARCHAR(50), headboxbg
VARCHAR(50));", $connection);
```

Tietokantaan lisätään myös taulu tutoriaalin tarvitsemille html-tageille,

kuten html-tiedoston alussa ja lopussa tarvittavat pakolliset tagit. Tutoriaalia luodessa luodaan tietokantaan automaattisesti myös pääsivun taulu. Pääsivun taulun nimi on page1 ja se pitää sisällään sivun otsikon.

```
mysql_query("CREATE TABLE html_tags(start_tags
VARCHAR(1000), end_tags VARCHAR(400), t_title VARCHAR(100),
footer_start VARCHAR(400), footer_end VARCHAR(400));",
$conection);
mysql_query("INSERT INTO html_tags(start_tags, end_tags,
t_title, footer_start, footer_end) VALUES('" . $start_tags
. "','" . $end_tags . "','" . $tutname . "','" . $foot-
er_start . "','" . $footer_end . "')");
mysql_query("CREATE TABLE page1(header VARCHAR(200));",
$conection);
mysql_query("INSERT INTO page1(header) VALUES('" . $tutname
. "')");
```

Kun Php-koodi on suorittanut tarvittavat MySql-komennot, luodaan tutoriaalille kansiorakenne. Kaikki luotavat tutoriaalit tehdään TutMaker-ohjelman juuressa olevaan kansioon Tutorials. Jokainen luotava tutoriaali saa oman kansionsa omalla nimellään. Tutoriaalikansion alikansioiksi luodaan images ja videos, jotka ovat varattu käyttäjän omille kuva- ja videotiedostoille. Kansioille asetetaan käyttöoikeudeksi 0777 eli oikeudet lukea, kirjoittaa ja suorittaa.

```
mkdir("./tutorials/" . $tutname, 0777);
mkdir("./tutorials/" . $tutname . "/images", 0777);
mkdir("./tutorials/" . $tutname . "/videos", 0777);
```

Kansiorakenteen luomisen jälkeen php-koodi luo tutoriaalisivuston tarvitsemat tiedostot. Ensin luodaan tutoriaalin tyylitiedosto, eli css-tiedosto. Tyylitiedoston lisäksi luodaan myös pää- ja navigointisivu. Kaikki kyseiset tiedostot luodaan php:n fopen-komennolla ja niihin kirjoitetaan fwrite-komennolla. Lopuksi tiedostot suljetaan fclose-komennolla.

```
$cssfile = "./tutorials/" . $tutname . "/" . $tutname .
"_style.css";
$fh = fopen($cssfile, 'w') or die("can't open file");
fwrite($fh, "\r\n");
fclose($fh);
```

Php-tiedoston lopussa suoritetaan iframen päivitys ja *opentut*-javascript funktio. *Opentut*-funktiossa tunnistetaan painettu painike ja suoritetaan sen mukainen koodi.

```
if(sender == "tut_new")
{
    dynamic_span_obj.innerHTML = "";
    createButtons("tut_new", "index", pagefile);
}
```

Lause tyhjentää käyttöliittymän sivuvalikon ja suorittaa *createButtons*-funktion, joka lisää sivuvalikkoon index-sivun.

5.1.2 Tutoriaalin avaus

Tutoriaalia avatessa käyttäjän tulee syöttää tekstikenttään avattavan tutoriaalin nimi tai valita avattava tutoriaali alasvetovalikosta. Tämän jälkeen käyttäjän pitää painaa ”Avaa”-painiketta (ks. Kuva 7). Painallus suorittaa tutoriaalin luonnin tapaan *tutorial_ctrl* Javascript-funktion, joka lähettää avattavan tutoriaalin nimen ja painetun painikkeen id:n eteenpäin *tutorial_ctrl.php* tiedostolle. Php-tiedostossa painettu painike tunnistetaan tutoriaalin avaus painikkeeksi. else if -rakenteen sisällä valitaan käyttäjän nimeämä tietokanta, lasketaan sivujen ja elementtien määrä sekä haetaan tutoriaalin tyyliattribuutit.

```
mysql_select_db($tutname, $connection) or
die(mysql_error());

//Haetaan sivujen maara
$page_result = mysql_query("SHOW TABLES LIKE 'page%'",
$connection);
$num_of_pages = mysql_num_rows($page_result);

//Haetaan elementtien maara
$element_result = mysql_query("SHOW TABLES LIKE
'element1%', $connection);
$num_of_elements = mysql_num_rows($element_result);

//Haetaan tyyli
$styles = mysql_query("SELECT * FROM style");

while ($row = mysql_fetch_array($styles)) {
    $bgcolor_a = $row["bgcolor_a"];
    $bgcolor_b = $row["bgcolor_b"];
    $bgimg_a = $row["bgimg_a"];
    $bgimg_b = $row["bgimg_b"];
    $sitewidth = $row["sitewidth"];
    $brdroption = $row["brdroption"];
    $brdrwidth = $row["brdrwidth"];
    $brdrcolor = $row["brdrcolor"];
    $fontoption = $row["fontoption"];
    $fontsize = $row["fontsize"];
    $fontcolor = $row["fontcolor"];
    $textboxbg = $row["textboxbg"];
    $headoption = $row["headoption"];
    $headsize = $row["headsize"];
    $headcolor = $row["headcolor"];
    $headboxbg = $row["headboxbg"];
}
```

Lopuksi suoritetaan pääkäyttöliittymän iframen päivitys ja *Opentut*-funktio, kuten tutoriaalin luonnin yhteydessä. Tällä kertaan *Opentut*-funktion sisällä painettu painike tunnistetaan tutoriaalin avaus painikkeeksi. Funktion sisällä suoritetaan pääkäyttöliittymän iframen päivitys muuttujalla, joka pitää sisällään tutoriaalin polun ja lisää siihen pääsivun tiedostonimen ja päätteen. Seuraavaksi tyhjennetään sivupainike- ja elementttilista

```
refreshIframe(pagefile+"index.html");
dynamic_span_obj.innerHTML = "";
clearOptions();
```

Tämän jälkeen luodaan avattavan tutoriaalin sivupainikkeet for-silmukalla, jota suoritetaan niin monta kertaa kun tutoriaalissa on sivuja. Silmukan sisällä on ehtolause, jolla määritellään onko sivu pääsivu vai normaalisivu. Pääsivun painike saa tekstikseen Index ja muut painikkeet sivu+silmukan kierrosnumero. Luotaviin painikkeisiin lisätään myös javascript *onClick* -funktio, joka suoritetaan kun painiketta painetaan.

```
for(var i = 1 ; i<=page_amount ; i++){
    if(i == 1){
        dynamic_span_obj.innerHTML += "<input
        type='button' value='index' id='"+i+"'
        onclick='javascript:btnFunc(&quot;./tutorials/" +
        document.getElementById('tutorialname').value +
        "/index.html&quot;);' /><br />";
    }else{
        dynamic_span_obj.innerHTML += "<input
        type='button' value='sivu"+i+"' id='"+i+"'
        onclick='javascript:btnFunc(&quot;"+pagefile+"pag
        e"+i+".html&quot;);' /><br />";
    }
}
```

Painikkeiden luonnin jälkeen täytetään elementtien alasvetovalikko suorittamalla *populateOptions*-funktio. Funktio ottaa sisäänsä avattavan tutoriaalin elementtien lukumäärän ja painetun painikkeen id:n, joka on tässä tapauksessa dummy-muuttuja "deletetxt".

```
populateOptions(elem_amount, "deletetxt");
```

PopulateOptions-funktion sisällä on for-silmukkarakenne. Silmukan sisällä suoritetaan funktio, joka lisää alasvetovalikkoon uuden valinnan. Silmukkaa suoritetaan yhtä monta kertaa kuin avattavassa tutoriaalissa on elementtejä.

```
function populateOptions(amount, sender)
{
    for(var i = 1; i <= amount ; i++){
        if(sender == "deletetxt"){
            appendOptionLast(i, "addtxt");
        }else if(sender == "deletetic"){
            appendOptionLast(i, "addpic");
        }else if(sender == "deletevid"){
            appendOptionLast(i, "addvid");
        }
    }
}
```

Lopuksi tutoriaalin tyylikenttiin asetetaan arvoiksi avattavan tutoriaalin tyyliasetukset.

```
document.getElementById('bgcolor_a').value = bgcolor_a;
document.getElementById('bgcolor_b').value = bgcolor_b;
document.getElementById('bgimg_a').value = bgimg_a;
document.getElementById('bgimg_b').value = bgimg_b;
document.getElementById('sitewidth').value = sitewidth;
document.getElementById('brdr_dropdown').value = rdoption;
document.getElementById('brdrwidth').value = brdrwidth;
document.getElementById('brdrcolor').value = brdrcolor;
document.getElementById('font_dropdown').value = ontoption;
document.getElementById('fontsize').value = fontsize;
document.getElementById('fontcolor').value = fontcolor;
document.getElementById('textboxbg').value = textboxbg;
document.getElementById('txtheadfont').value = headoption;
document.getElementById('fontsize_head').value = headsize;
document.getElementById('fontcolor_head').value = eadcolor;
document.getElementById('bgcolor_head').value = headboxbg;
    }
}
```

5.1.3 Tutoriaalin poisto

Poistaakseen aiemmin luodun tutoriaalin tulee käyttäjän syöttää poistettavan tutoriaalin nimi tekstikenttään. Poista-painikkeen painallus suorittaa MySQL-komennon, joka poistaa tutoriaalin tietokannan (ks. Kuva 6).

```
mysql_query("DROP DATABASE " . $tutname, $connection);
```

Tutoriaalin poisto ei poista tutoriaalin kansioita eikä tiedostoja.

5.2 Sivujen hallinta

Kuva 8 Sivujen hallinta

Sivujen hallinta-osion kaikki painikkeet suorittavat Page_ctrl-Javascript funktio, joka ottaa sisäänsä painetun painikkeen id:n. Funktio sisällä painikkeen id ja sivun otsikko tallennetaan omiin muuttujiinsa. Tarvittavat tiedot lähetetään edelleen Attach_file-funktiolla page_ctrl.php-tiedostolle. Lähetettäviä tietoja ovat tutoriaalin nimi, sivun otsikko, painikkeen id ja sivun numero. Sivun numero saadaan pagenummer-funktiolla, jonka sisällä suoritetaan regular expression toiminto. Regular expression –toiminnoilla poimitaan sivun numero avoimena olevan tutoriaalisivun tiedostonimestä.

```
function page_ctrl(btnId){
```

```

        var sender = btnId;
        var newpage_topic = document.getElementById("page_topic_field").value;

        attach_file('page_ctrl.php?tutname=' + document.getElementById('tutorialname').value + '&header=' + newpage_topic + '&sender=' + sender + '&page=' + pagenumber());
    }

    function pagenumber(){
        var contentfile =
        ""+document.getElementById('content').src;
        var filenum;
        var regex1 = /page\d+/i;
        var regex2 = /\d+/i;
        var file = ""+contentfile.match(regex1);

        if(contentfile.match(regex1)){
            filenum = ""+file.match(regex2);
        }else{
            filenum = 1;
        }

        return filenum;
    }

```

Page_ctrl.php tiedostossa painettu painike tunnistetaan id:n avulla. Painikkeiden id:t ovat: page_add, page_topic, page_del ja refresh eli sivun lisäys, sivun otsikon vaihto, sivun poisto ja iframen päivitys. Painikkeen id määrittää suoritettavan else if –rakenteen osion.

```

if ($sender == "page_add"), else if ($sender ==
"page_topic") ja else if ($sender == "page_del")

```

Page_ctrl.php tiedoston lopussa suoritetaan neljä javascript-komentoa. Ensimmäinen komento tallentaa esikatseluikkunan iframen muuttujaan. Toinen komento muuttaa content-muuttujan src-attribuutin, joka määrittää avattavan sisällön polun. Kolmas rivi suorittaa funktion, joka tyhjentää alavetovalikon. Viimeinen komento suorittaa createButtons-funktion, jota käytetään sivupainikkeiden luomiseen. Funktio ottaa sisäänsä painetun painikkeen id:n, sivujen lukumäärän ja sivun polun.

```

content_obj = document.getElementById('content');
content_obj.src = "<?php echo $pagefile; ?>";
clearOptions();
createButtons("<?php echo $sender; ?>", "<?php echo
$num_of_tables; ?>", "<?php echo $pagefile; ?>");

```

5.2.1 Sivun luonti

Uutta sivua luodessaan käyttäjä voi syöttää sivulle haluamansa otsikon. Luo uusi sivu –painike suorittaa page_ctrl-javascript-funktion, joka lähettää sivun otsikon edelleen page_ctrl.php-tiedostolle. Page_ctrl.php-tiedostossa uudelle sivulle luodaan oma taulu tietokantaan. Taulun kentiksi luodaan id ja otsikko (ks. Kuva 7).

```
mysql_query("CREATE TABLE page" . $num_of_tables . "(id INT
NOT NULL AUTO_INCREMENT, PRIMARY KEY(id), header
VARCHAR(200))", $connection) or die(mysql_error());
```

Seuraavaksi sivun taulun otsikkokenttään syötetään käyttäjän haluama otsikko.

```
mysql_query("INSERT INTO page" . $num_of_tables . "(header)
VALUES('<div id=\"page_header\">" . $page_topic .
"</div>')", $connection);
```

Tämän jälkeen tutoriaalin html-sivujen navigointi tulee päivittää. Päivitys tapahtuu kirjoittamalla navigoinnin html-sivuun, joka ladataan tutoriaalin sivujen alalaidassa olevaan iframeen. Navigoinnin sivupainikkeita saadaan oikea määrä for-silmukalla, jota toistetaan yhtä monta kertaa kun tietokannassa on sivutauluja. Silmukan sisällä on ehtolause, jolla saadaan silmukan ensimmäinen kierros tekemään linkki, joka johtaa tutoriaalin index.html-sivuun. Loput sivulinkit luodaan kierroksen järjestysnumeron mukaan.

```
for ($i = 1; $i <= $num_of_tables; $i++) {
    if ($i != 1) {
        fwrite($fh, "<a href=\"page" . $i . ".html\" tar-
get=\"_top\">" . $i . "</a>\r\n");
    } else {
        fwrite($fh, "<a href=\"index.html\" tar-
get=\"_top\">1</a>\r\n");
    }
}
```

5.2.2 Sivun otsikon vaihto

Käyttäjä voi vaihtaa valitsemansa sivun otsikon (ks. Kuva 8). Otsikon vaihtotoiminto suorittaa muiden toimintojen tapaan page_ctrl-funktion, jonka kautta suoritetaan page_ctrl.php-tiedoston osio, joka käsittelee sivun otsikon vaihtoon liittyvät toimenpiteet. Uusi otsikko tallennetaan tietokantaan ja sen jälkeen suoritetaan clsDrawPage-luokka, joka kirjoittaa sivun html-tiedoston.

```
mysql_query("UPDATE page" . $selected . " SET header='<div
id=\"page_header\">" . $page_topic . "</div>'", $connec-
tion) or die(mysql_error());
```

```
$num_of_elements = $createpage->drawpage($utname, $select-
ed);
```

5.2.3 Sivun poisto

Käyttäjä voi poistaa valitun sivun painamalla poista sivu -painiketta (ks. Kuva 8). Painikkeen painaminen suorittaa muiden toimintojen tapaan page_ctrl-javascript funktion. Page_ctrl.php-tiedostossa suoritetaan mysql-komento, joka poistaa valitun sivun taulun tietokannasta.

```
mysql_query("DROP TABLE page" . $selected, $connection) or
die(mysql_error());
```

Sivun taulun poistamisen jälkeen poistetaan sivun elementit. Ensin lasketaan elementtien kokonaismäärä. Jos sivulla on elementtejä, suoritetaan for-silmukka. For-silmukan sisällä suoritetaan Mysql-lause, jolla poistetaan sivukohtaiset elementit.

```
$resultg = mysql_query("SHOW TABLES LIKE 'element" . $se-
lected . "_"');
$num_of_elements = mysql_num_rows($resultg);

if ($num_of_elements > 0) {
    for ($k = 1; $k <= $num_of_elements; $k++) {
        mysql_query("DROP TABLE element" . $selected .
            "_" . $k, $connection) or die(mysql_error());
    }
}
```

Seuraavaksi lasketaan poiston jälkeinen sivumäärä ja muutetaan poistetun sivun jälkeisten sivutaulujen nimet vastaamaan niiden uusia järjestysnumeroita. Myös sivukohtaiset elementit nimetään vastaamaan uusia sivunumeroita. Taulujen uudelleennimeäminen tapahtuu for-silmukalla, jonka sisällä suoritetaan mysql-komentoja. For-silmukkaa suoritetaan niin monta kertaa kuin tutoriaalissa on sivuja, mutta uudelleennimeäminen alkaa vasta, kun for-silmukan kierrosten arvo on suurempi kuin poistetun sivun järjestysnumero. Kyseinen ehto saadaan aikaan if-ehtolauseella, jonka toteutuessa suoritetaan mysql:n taulun uudelleennimeämiskomento. Ehtolauseen sisällä suoritetaan toinen ehtolause, jolla tarkistetaan onko kyseisellä sivulla elementtejä. Jos sivulla on elementtejä, nimetään nekin uudelleen vastaamaan uusia sivunumeroita.

```
$resultc = mysql_query("SHOW TABLES LIKE 'page%'");
$num_of_tables = mysql_num_rows($resultc);

for ($i = 1; $i <= $num_of_tables; $i++) {
    $ik = $i + 1;

    if ($i >= $selected){

        mysql_query("RENAME TABLE page" . $ik . " TO
            page" . $i, $connection);

        $resultv = mysql_query("SHOW TABLES LIKE
            'element" . $ik . "_"');
        $num_of_elements = mysql_num_rows($resultv);

        if ($num_of_elements != 0) {

            for ($k = 1; $k <= $num_of_elements; $k++){
                mysql_query("RENAME TABLE element" . $ik . "_"
                    . $k . " TO element" . $i . "_" . $k,
                    $connection);
            }
        }
    }
}
```

```
}
```

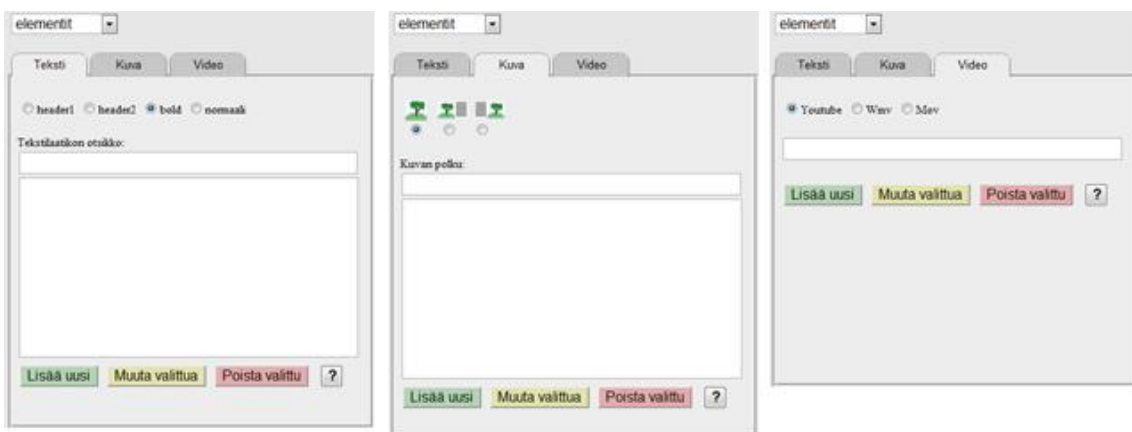
Tietokannan taulujen muutosten jälkeen muutetaan sivujen html-tiedostot vastaamaan uutta sivujärjestystä. html-sivujen nimimuutokset suoritetaan tuttuun tapaan for-silmukalla.

```
$pagefile = "../tutorials/" . $tutname . "/page" . $selected  
            . ".html";  
            unlink($pagefile);  
  
            for ($i = 1; $i <= $num_of_tables; $i++) {  
                $ik = $i + 1;  
  
                if ($i >= $selected) {  
                    rename("../tutorials/" . $tutname . "/page" . $ik  
                        . ".html", "../tutorials/" . $tutname . "/page" .  
                        $i . ".html");  
                }  
            }  
        }  
    }
```

Lopuksi päivitetään tutoriaalın navigointisivu. Päivitys tapahtuu for-silmukalla, jonka avulla luodaan navigointisivun sivulinkit.

```
for ($i = 1; $i <= $num_of_pages; $i++) {  
    if ($i != 1) {  
        fwrite($fh, "<a href=\"page\" . $i . \".html\" tar-  
            get=\"_top\">" . $i . "</a>\r\n");  
    } else {  
        fwrite($fh, "<a href=\"index.html\" tar-  
            get=\"_top\">1</a>\r\n");  
    }  
}
```

5.3 Teksti-, kuva- ja videotyökalut



Kuva 9 Teksti-, kuva- ja videotyökalut

Kaikki elementtien hallintaan liittyvät toiminnot näkyvät kuvassa 9. Ne suorittavat kukin omalle toiminnallisuudelleen luodut JavaScript-funktiot. Tekstityökalu suorittaa `textbox_ctrl`-funktion, kuvatyökalu `picbox_ctrl`-funktion ja videotyökalu `vidbox_ctrl`-funktion. Jokainen funktio ottaa sisäänsä käyttäjän painaman painikkeen nimen, painikkeiden nimet ovat:

add_elem, alter_elem ja delete_elem eli elementin lisäys, muokkaus ja poisto.

Kyseisten funktioiden sisällä toimitaan käyttäjän valitseman radio-painikkeen valinnan mukaisesti. Radio-painikkeita ovat esimerkiksi tekstityökalun otsikon muotoilun valinta. Valittu radio-painike saadaan selville seuraavalla JavaScript-koodilla:

```
for (i=0 ; i < document.radios.radiobtn.length ; i++){
    if (document.radios.radiobtn[i].checked == true){
        selected_radio = i;
    }
}
```

Tämän jälkeen suoritetaan if-ehdolauserakenne, josta suoritetaan radio-valinnan mukainen osio. Alla oleva esimerkki on textbox_ctrl-funktion sisältämästä ehdolauseesta. Tekstilaatikon otsikon tyyliksi valitaan käyttäjän valinnan mukaisesti joko lihavoitu, otsikko 1, otsikko 2 tai normaali teksti. Jokainen otsikkotyyli koostuu aloittavasta ja päättävästä html-tagista, jotka tallennetaan omiin muuttujiinsa.

```
if(selected_radio == 2){
    htypea = "<b>";
    htypeb = "</b>";
}else if(selected_radio == 0){
    htypea = "<h1>";
    htypeb = "</h1>";
}else if(selected_radio == 1){
    htypea = "<h2>";
    htypeb = "</h2>";
}else if(selected_radio == 3){
    htypea = "";
    htypeb = "";
}
```

Tämän jälkeen valitun radio-painikkeen mukaiset html-tagit muotoillaan tekstilaatikon rakenteen muodostavien div ja span -tagien sisään. Muotoillut merkkijonot tallennetaan kolmeen eri muuttujaan.

```
tags_a = "<div class=\"textbox_div\">\r\n<div
class=\"txtbox_header\">\r\n<span
class=\"header_span\">"+htypea;
tags_b = htypeb+"</span>\r\n</div>\r\n<div
class=\"txtbox_text\">\r\n<span class=\"text_span\">\r\n";
tags_c = "</span>\r\n</div>\r\n</div>\r\n";
```

Muista elementtityypeistä poiketen tekstilaatikon alityyliksi määritetään aina "textbox", koska tekstilaatikon html-tagien suorittamisjärjestys on aina sama. Muissa elementeissä, kuten kuvaelementeissä html-tagien suorittamisjärjestys määräytyy alityylin mukaisesti.

Lopuksi suoritetaan attach_file-funktio, kuten muidenkin työkalujen funktioissa. Kaikki elementtityökalut suorittavat element_ctrl-funktion.

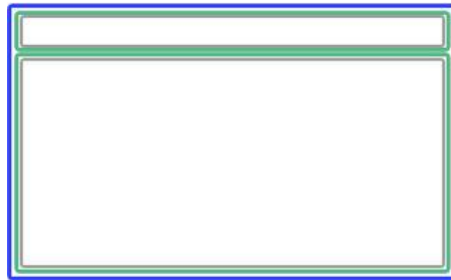
5.3.1 Elementtien ominaisuudet

Tekstityökalulla käyttäjä voi lisätä tutoriaaliin tekstilaatikoita, joihin hän voi itse määrittää otsikon ja sisältötekstin. Käyttäjä voi valita otsikolle haluamansa tyylin. Tekstilaatikon sisältötekstin kirjoituskenttään voidaan kirjoittaa html-koodia. Tämä mahdollistaa esimerkiksi taulujen ja listojen tekemisen (ks. Kuva 9).

Kuvatyökalulla käyttäjä voi lisätä sivulle kuvan. Vaihtoehtoina kuvanlisäykselle ovat pelkkä kuva, kuva ja tekstilaatikko kuvan oikealla tai vasemmalla puolella. Kuvanlisäystyökalu sisältää myös tekstinsyöttökentän johon kuvateksti voidaan syöttää (ks. Kuva 9).

Videotyökalulla voidaan lisätä tutoriaaliin kolmea eri formaattista videomateriaalia. Eri formaatteja ovat: Youtube, Windows Media Video ja Quick Time Video. Youtube-videota lisätessä käyttäjän tulee syöttää tekstikenttään vain videon koodi, joka löytyy videon Youtube-sivulla selaimen osoitekentästä. Muissa tapauksissa käyttäjän tarvitsee syöttää tekstikenttään vain videon polku (ks. Kuva 9).

Jokainen elementtityökalu luo html-koodia omalla tavallaan, mutta edistynyt käyttäjä voi käyttää esimerkiksi vain tekstityökalua muidenkin elementtien lisäykseen. Tämä johtuu siitä, että tekstikenttään voi syöttää vapaasti html-koodia, joka luetaan osana luotua tutoriaalin sivua. Hakkerointiyritysten välttämiseksi tekstikenttien sisältö tarkistetaan Regularexpression-menetelmällä, jolla tekstistä karsitaan mahdolliset JavaScript ja Php -koodit.



Kuva 10 Tekstilaatikon rakenne

Tekstilaatikat koostuvat pää div-kehyksestä, jonka sisällä on omat div-kehyksensä tekstilaatikon otsikolle ja tekstile. Otsikon ja tekstin div-kehysten sisällä on omat span-kehyksensä, joita käytetään tekstin muotoiluun (ks. Kuva 10).



Kuva 11 Kuvalaatikon rakenne

Kuvatekstillinen kuvalaatikko koostuu taulusta, joka on jaettu kahteen osaan. Taulun toiseen osaan tulee kuva ja toiseen teksti. Tekstin puoleisessa sarakkeessa on div-kehys, jonka sisälle teksti tulee. Teksti ja kuvapuolen järjestys riippuu käyttäjän valinnasta. Ilman kuvatekstiä oleva kuva lisätään sivulle ilman minkäänlaista taulurakennetta (ks. Kuva 11).

Videolaatikoilla ei myöskään ole erillistä taulurakennetta. Windows Media Video ja Quick Time –formaatin videot lisätään ”<object>” ja ”<embed>” html-komennoilla. Youtube-video ladataan omaan iframe-tauluunsa. Alla esimerkki Windows Media Videon lisäyskoodista.

```
<object id="MediaPlayer" width=320 height=286
classid="CLSID:22D6f312-B0F6-11D0-94AB-0080C74C7E95"
standby="Loading Windows Media Player components..."
type="application/x-oleobject" code-
base="http://activex.microsoft.com/activex/controls/mplayer
/en/nsmp2inf.cab#Version=6,4,7,1112">

<param name="filename" value=" videos/video.wmv ">
<param name="Showcontrols" value="True">
<param name="autoStart" value="True">

<embed type="application/x-mplayer2" src="videos/video.wmv"
name="MediaPlayer" width=320 height=240></embed>
</object>
```

Esimerkki Youtube-videon lisäyskoodista:

```
<iframe width="420" height="315"
src="http://www.youtube.com/embed/qs3HBZC-AY8"
frameborder="0" allowfullscreen></iframe>
```

5.3.2 Elementin lisäys

Elementin lisäys suorittaa kaksi Mysql-komentoa, joilla luodaan tietokantaan taulu ja kirjoitetaan luotuun tauluun käyttäjän syöttämät tiedot. Syötettävään tietoon kuuluu muun muassa: elementin tyyppi, elementin tarvitsemat html-käskyt, otsikko ja teksti.

Tietokantatoimenpiteiden jälkeen suoritetaan clsDrawPage.php-ulkoinen luokka, joka kirjoittaa muuttuneet tiedot tutoriaalın sivun html-tiedostoon.

```
mysql_query("CREATE TABLE element" . $selected . " " .
$num_of_tablesb . "(type VARCHAR(50), subtype VARCHAR(50),
tags_a VARCHAR(400), tags_b VARCHAR(400), tags_c
VARCHAR(400), header VARCHAR(500), text_value
VARCHAR(800))", $connection) or die(mysql_error());

mysql_query("INSERT INTO element" . $selected . " " .
$num_of_tablesb . "(type, subtype, tags_a, tags_b, tags_c,
header, text_value) VALUES('textbox','" . $subtype . "','"
. $tags_a . "','" . $tags_b . "','" . $tags_c . "','" .
$textbox_topic . "','" . $text . "')" , $connection);
```

```
$num_of_elements = $createpage->drawpage($tutname, $selected);
```

5.3.3 Elementin tietojen muuttaminen

Käyttäjä voi valita aiemmin lisätyn elementin alasvetovalikosta. Valinta suorittaa `change_textbox.php`-tiedoston, joka hakee tietokannasta valitun elementin otsikon ja tekstin. Haetut tiedot viedään `fillTextbox-javascript`-funktiolle, joka täyttää työkalulaatikon tekstikentät valitun elementin tiedoilla. Tiedot avataan sillä hetkellä valittuna olevan välilehden tekstikenttiin, riippumatta siitä minkä tyyppinen elementti on kyseessä.

Painamalla ”Muuta valittua” –painiketta käyttäjä voi tallentaa muutetut tiedot tietokantaan. Php-tiedostossa suoritetaan `update` Mysql-komento, jolla päivitetään elementin taulu uusilla tiedoilla.

```
mysql_query("UPDATE element" . $selected . "_" . $optionx .
" SET subtype='" . $subtype . "', tags_a='" . $tags_a . "',
tags_b='" . $tags_b . "', tags_c='" . $tags_c . "', head-
er='" . $textbox_topic . "', text_value='" . $text . "'",
$conection) or die(mysql_error());
```

```
$num_of_elements = $createpage->drawpage($tutname, $selected);
```

5.3.4 Elementin poisto

Käyttäjä voi poistaa aiemmin luodun elementin valitsemalla alasvetovalikosta haluamansa elementin ja painamalla ”Poista valittu” –painiketta. Elementin poistaminen suorittaa php-tiedostossa `drop mysql`-komennon, jolla elementin taulu poistetaan tietokannasta. Tämän jälkeen sivun elementtitaulut lasketaan uudestaan ja suoritetaan `for`-silmukka, jolla jäljellä olevien elementtitaulujen nimet muutetaan. Taulujen uudelleennimeämislogiikka toimii samaan tapaan kuin sivun poiston yhteydessä.

```
mysql_query("DROP TABLE element" . $selected . "_" .
$optionx, $conection) or die(mysql_error());

$resultc = mysql_query("SHOW TABLES LIKE 'element" . $se-
lected . "_%'");
$num_of_tables = mysql_num_rows($resultc);

for ($i = 1; $i <= $num_of_tables; $i++) {
    $ik = $i + 1;

    if ($i >= $optionx) {
        mysql_query("RENAME TABLE element" . $selected .
        "_" . $ik . " TO element" . $selected . "_" . $i,
        $conection);
    }
}

$num_of_elements = $createpage->drawpage($tutname, $se-
lected);
```

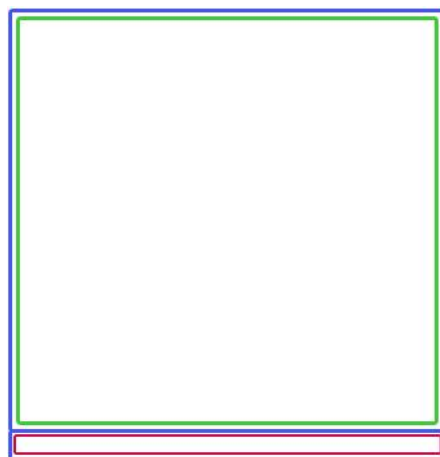
5.4 Sivuston tyylitiedoston asetukset



Kuva 12 Tutoriaalin tyylitiedoston hallinta ja väripaletti

Käyttäjä voi muokata tutoriaalisivuston tyyliasetuksia, kuten esimerkiksi tausta- ja fonttivärejä, sekä fonttikokoja. Painallus värinvalinta laatikkoon aukaisee jscolor JavaScript-työkalun, jolla voi helposti valita haluamansa värin (ks. Kuva 12). Valittu väri ilmoitetaan värikoodina. Painamalla ”Päivitä tyylitiedoston asetukset” –painiketta käyttäjä lähettää kenttien tiedot style_ctrl JavaScript-funktiolle, jossa arvot tallennetaan muuttujiin ja lähetetään edelleen style_ctrl.php-tiedostolle. style_ctrl.php-tiedostossa tiedot tallennetaan tietokantaan ja kirjoitetaan tutoriaalin css-tyylitiedostoon.

5.5 Luotu tutoriaalisivu



Kuva 13 Tutoriaalisivun perusrunko

Jokaiseen tutoriaalin html-tiedostoon kirjoitetaan html-kielellä koodi, joka luo sivun perusrunkon (ks. Kuva 13). Kuvassa sinisellä on sivun päätaulu, joka on keskitetty sivun keskelle. päätaulu on jaettu kahteen päällekkäiseen sarakkeeseen, joista ylempi sisältää ”<div>”-taulun, johon kaikki tutoriaalin sisältö lisätään. Päätaulun alempi sarake sisältää ”<iframe>”-kehyksen, jonka sisään ladataan sivuston navigointi.

6 YHTEENVETO JA POHDINTAA

6.1 Nettitutoriaaliohjelman suunnittelu ja toteutus

Opinnäytetyössä suunniteltiin ja toteutettiin nettitutoriaaliohjelma mainostoimistolle osaksi sen yrityksille tarjoamia verkkopalveluja. Nettitutoriaalia tarvitaan erilaisiin ohjeistuksiin, joita on aiemmin toteutettu paperiesittein ja pdf-formaatissa. Projektia ohjaavana kysymyksenä oli se, miten nettitutoriaaliohjelman tekninen ja visuaalinen toteutus yhdessä takaisivat sen monipuolisen käytettävyyden varsinkin administraattoreiden eli mainostoimiston ammattilaisten keskuudessa kuitenkin unohtamatta valmiin tutoriaalin päätekäyttäjiä erilaisissa yrityksissä ja niiden asiakaskunnassa.

Nettitutoriaaliohjelma suunniteltiin tukeutuen Jacob Nielsenin (1993) käytettävyysteoriaan, josta käytettävyyden ja hyödyllisyyden lisäksi tärkeimpinä kriteereinä toimivat opittavuus, tehokkuus, miellyttävyys, muistettavuus ja virheettömyys.

Nettitutoriaaliohjelma vastaa näihin kriteereihin helposti opittavalla ja miellyttävän ulkoasun omaavalla käyttöliittymällä. Käyttöliittymän selkeä ulkoasu tekee siitä myös erittäin helposti opittavan. Nettitutoriaaliohjelman täyttä tehokkuutta ei ole testattu johtuen oikeiden tutoriaalikohdeiden puutteesta. Ohjelman tehokkuutta olisi edistänyt esimerkiksi ominaisuus, jolla käyttäjä voisi vaihdella sivujen ja sisältöelementtien paikkoja. Tällä hetkellä se ei ole mahdollista, mutta logiikka kyseiselle toiminnolle on jo suunniteltu.

Täysi virheettömyys ei toteudu nettitutoriaaliohjelman käytettävyydessä. Virheitä saattaa tapahtua esimerkiksi esikatseluikkunan päivityksen yhteydessä, jolloin käyttäjä joutuu joko päivittämään ikkunan manuaalisesti tai avaamaan tutoriaalin uudestaan. Virhetilanteen sattuessa nettitutoriaaliohjelma ei ilmoita siitä käyttäjälle. MySql-virheen sattuessa virhe kyllä käsitellään ja MySql yhteys lopetetaan, mutta käyttäjä ei saa tapahtumasta tietoa. Käyttäjä ei esimerkiksi saa tietoa, jos hän yrittää luoda tutoriaalia nimellä, joka on jo tietokannassa.

6.2 Toimintaympäristö

Nettitutoriaaliohjelman testaus on suoritettu vain virtuaalisessa serveriympäristössä. Sovelluksen täydellinen testaus olisi tarvinnut oikean palvelimen, joka tukee PHP:ta. Nettitutoriaaliohjelman toiminnassa saattaa ilmetä ongelmia oikealla palvelimella. Käytännössä sovellus toimii nopeammin virtuaalipalvelimella, koska virtuaalipalvelin on samalla tietokoneella kuin käyttäjä. Oikealla palvelimella sovelluksen toiminta saattaa hidastua huomattavasti johtuen internetin viiveestä ja mahdollisesta ruuhkautumisesta.

Nettitutoriaaliohjelmaa ei myöskään ole testattu oikeiden asiakkaiden tarpeita vastaavilla tutoriaaleilla, joten tekninen suunnitteluprosessi

(engineering design) on jäänyt kesken. Koko sovellus on suunniteltu ja toteutettu yleisten ja tietyllä tavalla minimalististen web-sivustandardien mukaiseksi, mutta mainostoimiston arjessa asiakas saattaa tahtoa jotain, mihin nettitutoriaaliohjelma ei kykenekään vastaamaan. Kyseisiä asioita voivat olla esimerkiksi monimutkaiset ja tietynlaista asettelua vaativat graafiset elementit sekä yleisesti sisällön jatkomuotoilu.

6.3 Toteutuksessa ilmenneet ongelmat

Toteutuksen kannalta suurin ongelma oli sovelluksen ohjelmointiin käytettävä aika, koska en tehnyt nettitutoriaaliohjelmaa päätyökseksi. Suurimman osan ohjelmoinnista suoritin Preciksessä Mac-tietokoneella 3d-kuvien renderöinnin aikana ja viikonloppuisin Windows-tietokoneella kotona. Eri käyttöjärjestelmien takia ongelmaksi nousivat myös pienet erot internetselainten Mac ja Windows versioiden sivuntaitossa, joka johti eroavaisuuksiin käyttöjärjestelmästä riippuen. Sain kuitenkin lopulta karsittua lähes kaikki Safari, FireFox ja Google Chrome –internetselainten väliset ongelmat. En nähnyt tarpeelliseksi testata nettitutoriaaliohjelmaa Internet Explorerilla, sillä se on Windows-ympäristön selain ja Preciksessä käytetään pääasiassa FireFoxia.

6.4 Oppina itselle

Nettitutoriaaliohjelman suunnittelu- ja luontiprosessit olivat loistavaa oppia vastaavien ohjelmointiprojektien varalle, koska en ollut aikaisemmin luonut internetpohjaisia sovelluksia Php-kielellä. Staattisten internetsivujen tekeminen oli minulle ennestään tuttua, mutta dynaamisen kokonaisuuden luominen opetti minut käyttämään MySql-tietokantaa osana sivuston toimintaperiaatetta. Projektista oppi myös tiettyjä loogisia koodirakenteita, joita voi hyödyntää muissakin ohjelmointiprojekteissa.

6.5 Jatkotutkimus ja kehittäminen

Internetsivujen sisällöntuottoon tarkoitettuja sovelluksia löytyy paljon, mutta tutoriaalien luontiin pyhitettyjä sovelluksia ei nettihauilla löytynyt, ei myöskään tutkimus- tai kehitystöitä. Nettitutoriaaliohjelmalla luotavien tutoriaalien kaltaisia tutoriaaleja voidaan toki tehdä normaaleilla sisällöntuotto-sovelluksillakin. Normaaleihin sisällöntuotto-sovelluksiin verrattuna tässä opinnäytetyössä kehitetty nettitutoriaaliohjelma kuitenkin eroaa joiltakin osin. Esimerkiksi nettitutoriaaliohjelma luo jokaisesta tutoriaalista oman kokonaisuutensa, joka voidaan helposti jakaa kansiona asiakkaalle. Jatkotutkimuksella saataisiin selville asiakaskäyttäjien mielipiteitä ja parannusehdotuksia, jolloin nettitutoriaaliohjelmasta voitaisiin kehittää vielä laajemmin käyttökelpoisempi.

Jatkokehityksenä voisivat olla myös valmiit asiakaskohtaiset tutoriaalirungot, jotka mahdollistaisivat automaattisen sivumuotoilun ja värit, sekä sisältöelementtien paikat asiakkaan ennalta määäämien arvojen mukaisesti. Asiakaskohtaisilla tutoriaalirungoilla saataisiin jokaisesta asiakaskohtaisesta tutoriaalista yhtenevä. Myös virhetilanteista

ilmoittaminen tulisi lisätä, jotta käyttäjä tietäisi mahdollisista virhetilanteista ja siitä kuinka virheen voisi välttää.

Muita parannuksia voisi olla muun muassa parempi ja monipuolisempi tekstieditori, jolla voisi helposti lihavoida, kursivoida ja alleviivata, sekä lisätä linkkejä ilman html-koodin osaamista. Nettitutoriaali ohjelmasta puuttuu myös mahdollisuus ladata kuvia ja videoita palvelimelle. Kuvien latauksen yhteyteen sopisi automaattinen kuvien pienennys ja esikatselukuvien luonti.

LÄHTEET

Nielsen, Jakob (1993) Usability Engineering, San Francisco: Morgan Kaufmann.

(IHTE) Kaper s2007 Luento 6: ”Käytettävyyden hyötyjä ja arviointimenetelmiä.” Tampereen teknillinen yliopisto, Ihmiskeskeisen teknologian yksikkö
(<http://www.cs.tut.fi/~kaper/syksy07/luennot/S07L6.pdf>).

Wikipedia 1: Käytettävyys
(<http://fi.wikipedia.org/wiki/K%C3%A4ytett%C3%A4vyys>)

Wikipedia 2: Engineering Design
(http://en.wikipedia.org/wiki/Engineering_design_process)

Ohjelmoinnissa apuna käytetyt lähteet:

Rosebrock, Eric (2003) Introduction to PHP Classes (OOP)
(<http://www.devarticles.com/c/a/PHP/Introduction-to-PHP-Classes-OOP/>) 22.4.2003

Vaughn, Brian (2005) How Does This Work? – PHP
(<http://www.devshed.com/c/a/PHP/PHP-and-JavaScript-Pooling-Your-Resources/3/>) 20.9.2005

www.pantz.org (2007) MySQL Commands
(<http://www.pantz.org/software/mysql/mysqlcommands.html>) 25.07.2007

www.w3schools.com JavaScript-tutoriaalit
(<http://www.w3schools.com/js/>)

www.tuxradar.com (<http://www.tuxradar.com/practicalphp/4/0/0>)

www.php.net (<http://www.php.net/>)